

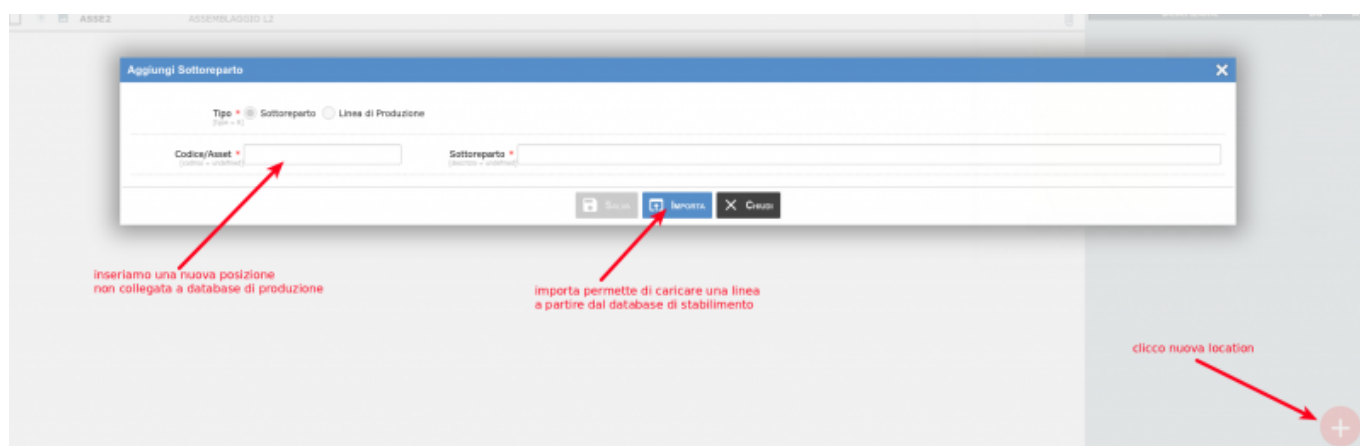
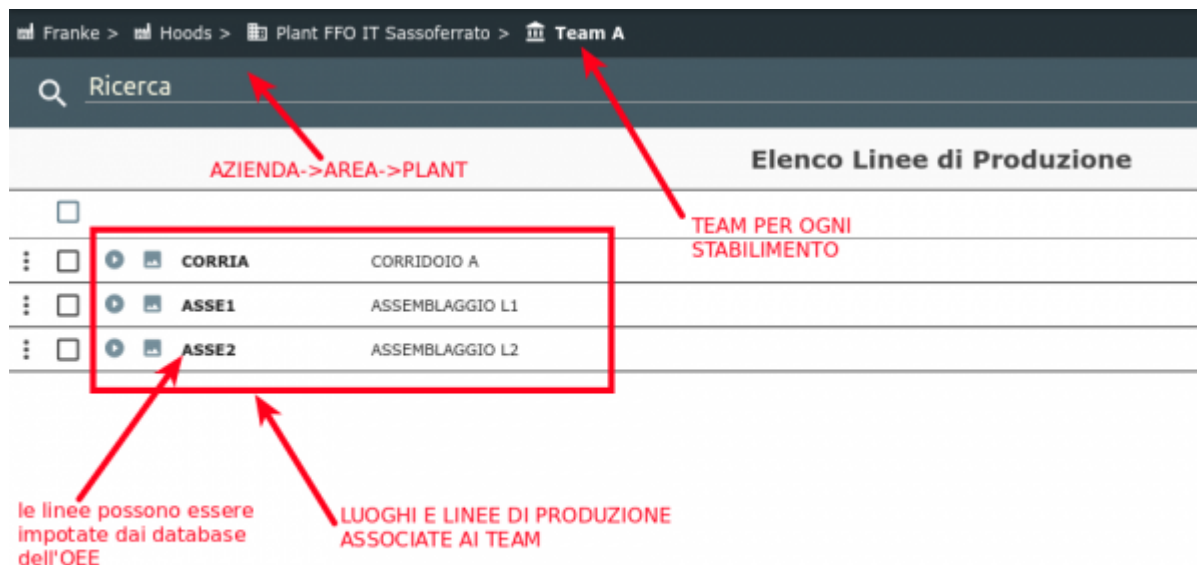
Digital Team Board and CIC Board

Di seguito il documento di specifica di Franke:

- Digital Team Board and CIC Board

La struttura delle location prevederà:

- Azienda → Area → Plant
- Team (Reparti) → Luoghi e linee di produzione



Al fine di importare le location da un database di un determinato stabilimento per aggiungerle ai relativi TEAM, si procede come segue:

1. Si seleziona la lista delle ubicazioni che si vogliono importare dallo stabilimento
2. Si richiama la setcentubi dell'agente centubi

Per il p.to 1 è necessario usare l'agentdb definito nella tabella dbubi in corrispondenza dello stabilimento (tiposet='S') relativo all'elemento selezionato. Così si può estrarre la lista di tutte le linee di produzione ordinate in modo alfabetico. Successivamente si può richiamare la seguente per inviare al centubi che copierà i dati:

Test Agenti

Opzioni agenti

Agente: Tell ▾

Request:

P1:

P2:

Nume:

Timeout:

Descrizione:

Opzioni utente

☐ Non chiedere conferma esecuzione ☐ Non mostrare risultato ☐ Mostra il risultato ir

Safety Pages

Selezionando un elemento della gerarchia a partire dal Team (Reparto) in poi viene abilitata una delle funzioni di inserimento dati presenti ai paragrafi:

- 1.2.2, 1.2.3, 1.2.4

	☒			CORRIA	CORRIDOIO A			
	☐			ASSE1	ASSEMBLAGGIO L1			
	☐			ASSE2	ASSEMBLAGGIO L2			

ognuna di queste icone (che verranno cambiate) permetterà l'accesso ad uno dei possibili inserimento dati

Strutture dati

Le strutture dati utilizzate sono quelle relativa al sistema EEM ovvero:

- objiter: ogni oggetto della safety è un objiter abbiamo i seguenti tipi di oggetti:
 - Near Miss (N)
 - Accident (A)

- Long time 5S (S)
- Long time TMP (J)
- Short time 5S/TMP (Y)
- Audit Checklist (P)
- project: ogni oggetto trova i dati di dettaglio nel project
- objteam: team interessato dall'attività

Per quanto riguarda oggetti di tipo N, A, S, J, Y viene considerato il project solo come definizione di eventi e attività da svolgere; un'attività è svolta quando l'objiter.state='A' (approvato/chiuso).

Mentre per quanto riguarda il tipo P, si dovrà generare la checklist (clista) e registrare l'esito del voto direttamente in curans.

Near Miss, Accident, Long time 5S, Long time TMP sono gestiti dalla pagina al paragrafo 1.2.4, mentre Short time 5S/TPM sono gestiti dalla pagina 1.2.2. I flussi operativi sono esattamente gli stessi, ma quando si crea un progetto con la 1.2.2 si dovrà assegnare anche un project.codequip.

Una struttura completa del database la troviamo ai seguenti paragrafi:

- [Objiter](#)
- [Checklist](#)
- [Project](#)

Near Miss, Accident, long time 5S e TPM, short time TPM

Il flusso generale della gestione dei suddetti objects è il seguente:

1. newobjiter (tipo=N,A,S,J,Y): crea il nuovo objiter e restituisce il **codice**
2. insert into project (codice,descrizio,codubi): inserisce in project con la descrizio ed il codubi (vedi sotto le query da svolgere)
3. initteam: inizializza il team, nel caso specifico individua "chi" dovrà eseguire l'eventuale contromisura
4. sendobjiter: pianifica l'objiter per lo svolgimento
5. approve: dichiara lo svolgimento eseguito dell'objiter

L'inserimento del progetto deve procedere svolgendo esattamente le seguenti query, sostituendo opportunamente il codice del progetto con il codice del objiter create (le tre query vanno fatte in un colpo solo):

```
INSERT INTO PROJECT (codice,descrizio,codubi) VALUES (8,'descrizione di
prova','000000098'); -- (alloca il progetto)
INSERT INTO PRJMATGAP (CODPRJ,CODGRP,CODAREA,CODPROC) VALUES (7,1,1,1); --
(necessario per la gestione delle checklist)
INSERT INTO PRJCAT (codprj,codcat) SELECT 7,category.codice FROM category; --
(necessario per la gestione delle checklist)
```

Le tabelle coinvolte dai movimenti sono le seguenti:

- objiter: contiene la tipologia ed i riferimenti principali
- project: contiene alcune informazioni di dettaglio (descrizio, codubi, altro...)
- objteam: contiene il team (chi) che deve svolgere; il team avrà objteam.tipo=A
- iter: contiene le date di previsto svolgimento (tmsend) ed infine di svolgimento (tmfeed)

L'objiter cambia stato in funzione delle operazioni:

- appena inserito è in objiter.state=N
- dopo la sendobjiter è in objiter.state=S

- dopo l'approve è in objiter.state=A (stato di chiusura dell'operazione)

Di seguito le chiamate da effettuare ad i rispettivi agenti:

- **newobjiter** (nel nume va il tm di creazione dell'oggetto, se 0 prende il tm attuale)

Opzioni agenti	
Agente	iter Ask ▼
Request	newobjiter
P1	{tipo:"A",codoper:"admin"} e
GetRec Req P1	newrec
GetRec Name P1	jsrec
GetRec Len P1	51 (ATTENZIONE: dimensione di ogni campo del rec,
P2	
Nume	0
Timeout	60000
Descrizione	

- **initteam**: nel caso di modifica del team è sufficiente reinviare il team con nume=1 (il team verrà cancellato e ripristinato, al limite si fa una select dell'owner e si invia anche l'owner; in tal caso si dovrà aggiungere un campo al rec con il nome "owners" e mettere il codoper opportuno)

Opzioni agenti	
Agente	iter Tell ▼
Request	initteam
P1	{"codice":6,"ateam":"admin"}
GetRec Req P1	newrec
GetRec Name P1	jsrec
GetRec Len P1	51 (ATTENZIONE:
P2	
Nume	
Timeout	60000
Descrizione	

- **sendobjiter** (nel nume va il tm di creazione dell'oggetto, se 0 prende il tm attuale)

Opzioni agenti	
Agente	iter Tell ▼
Request	sendobjiter
P1	{"codice":6,"sender":"admin"}
GetRec Req P1	newrec
GetRec Name P1	jsrec
GetRec Len P1	51 (ATTENZIONE:
P2	
Nume	
Timeout	60000
Descrizione	

- **approve** (nel nume va il tm di creazione dell'oggetto, se 0 prende il tm attuale)

Opzioni agenti

Agente	iter	Tell
Request	approve	
P1	<input type="text" value='{"codice":6,"codoper":"admin"}'/>	
GetRec Req P1	newrec	
GetRec Name P1	jsrec	
GetRec Len P1	51	(ATTENZIONE: dimensione)
P2		
Nume		
Timeout	60000	
Descrizione		

La seguente immagine illustra il flusso dall'inizializzazione del team alla chiusura dell'oggetto nella grafica proposta da Franke.

[illegible]

Mentre i short time TPM/5S sono gestiti dalla seguente ove è indicato il flusso:

TPM / 5S			check: iter.tmsend < tm corrente						
Frage			Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Sonntag
Team Board	1	xxxxxx							
	2	Linie 1 cabine	Simon			Simon			
Maschine 5S	3	Linie 2 boden			Simon				
	4	Linie 5 markierungen		Bladimir					
Zone 5S	5	xxxxxx				Bladimir			
	6	xxxxxx					Hans		
	7	xxxxxx					Fritz		
	8	xxxxxx					Hugo		
WIP	11	xxxxxx				Hans			
Env.	12	xxxxxx							
Tagliche TPM Fragen	13	xxxxxx							
	14	xxxxxx							
	15	xxxxxx							
	16	xxxxxx							
	17	xxxxxx							
	18	xxxxxx							
	19	xxxxxx							
	20	xxxxxx							
	21	xxxxxx							
	22	xxxxxx							
	23	xxxxxx							

Short time TPM/5S REJECT

Dallo schema del short time TPM/5S è evidente come in corrispondenza di una determinata riga, vi sia una ripianificazione di una attività.

TPM / 5S			Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Sonntag
Team Board	1	xxxxxx							
	2	Linie 1 cabine	Simon			Simon			
Maschine 5S	3	Linie 2 boden			Simon				
	4	Linie 5 markierungen		Bladimir					
	5	xxxxxx				Bladimir			

In questo caso specifico, l'operatore deve avere la possibilità di eseguire i seguenti passi:

1. reject: una reject in corrispondenza della ITER che non è stata svolta
2. sendobjiter: una sendobjiter per creare la nuova ITER da svolgere

La prima sendobjiter crea il primo iter con lo stato = 'N' (vedi sotto).

codice	codobj	codoper	tmsent	tmfeed	comment	state	datasend	datafeed
4	21	NULL	1604485559	0		N	20201104	

A partire dal calendario possiamo prevedere le seguenti funzionalità:

- conferma svolgimento (approve)
- attività non svolta (reject)

La conferma corrisponde alla chiamata **approve** già vista sopra, mentre la reject è la seguente.

Test Agenti

Opzioni agenti

Agente: iter Tell

Request: reject

P1: [{"codice":21,"codoper":"admin","comment":"Ho dovuto ripianificare"}] esempio r

GetRec Req P1: newrec

GetRec Name P1: jsrc

GetRec Len P1: 51 (ATTENZIONE: dimensione di ogni campo del rec, non di t

P2: Con

Nume:

Timeout: 60000

Descrizione:

Funziona esattamente come la approve ma l'effetto su iter è il seguente.

- su iter

codice	codobj	codoper	tmsent	tmfeed	comment	state	datasend	datafeed
4	21	admin	1604485559	1604485769	Ho dovuto ripianificare	R	20201104	20201104

tmfeed assegnato

state=R con commento

- su objiter

codice	codoper	tmcre	datacre	state	tmcanc	tipocanc	datacanc	codpar	napprov	hh	min	sec	dayofweek	dayofyear	dayofmonth	month	year	weekofyear	oracre
21	admin	1604485458	20201104	R	0	Y	NULL		1	11	24	18	3	308	4	11	2020		45 11:24:18

anche objiter nello stato R

Se rigettiamo una attività, abbiamo la possibilità di riprogrammarla richiamando nuovamente la sendobjiter con una nuova data. Effettuando questa operazione otteniamo quanto segue:

- su iter

codice	codobj	codoper	tmsent	tmfeed	comment	state	datasend	datafeed
4	21	admin	1604485559	1604485769	Ho dovuto ripianificare	R	20201104	20201104
5	21	NULL	1604486150	0		N	20201104	

nuova iter nello stato N

- su objiter

codice	codoper	tmcre	datacre	state	tmcanc	tipo	datacanc
21	admin	1604485458	20201104	S	0	Y	

objiter torna
nello stato S

Si può iterare questo flusso finchè si rigetta definitivamente (objiter resta nello stato R) oppure si approva (objiter nello stato A).

Interrogando la tabella iter si può ricavare una sorta di calendario delle varie pianificazioni effettuate.

Short time TPM/5S Eventi periodici

E' possibile pianificare eventi con una cadenza periodica a tal fine si stabilisce:

- data di inizio pianificazione (project.tmstart)
- data di fine pianificazione (project.tmend)
- tipo di periodo (project.tipoper): W=settimanale, D=giornaliero
- durata del periodo in unita' di tipoper (project.n): ogni 2 giorni $\Rightarrow$ tipoper=D, n=2

In questo caso si dovranno utilizzare apposite chiamate per gestire gli objiter periodici che sono le seguenti:

- pianificazione: sendobjiterperiodic
- annulla piano: undosendobjiterperiodic
- approve di una iter: approveiterperiodic
- reject di una iter: rejectiterperiodic
- undo di una iter: undoiter (con nume=1)

Di seguito la sendobjiterperiodic.

Test Agenti

Opzioni agenti

Agente: iter Tell: ▼

Request: sendobjiterperiodic

P1: [{"codice":40,"sender":"admin","tmstart":1605430800,"tmend":1607158800,"tipo":"W",n:1}] esempio non

GetRec Reg P1: newrec

GetRec Name P1: jarec

GetRec Len P1: 51 (ATTENZIONE: dimensione di ogni campo del rec, non di tut

P2: Conti

Nome:

Timeout: 60000

Descrizione:

come la sendobjiter

dati per la pianificazione periodica presi da project

Di seguito la undosendobjiterperiodic.

Test Agenti

Opzioni agenti

Agente: iter Tell ▼

Request: undosendobjiterperiodic

P1: {"codice":40}

GetRec Req P1: newrec

GetRec Name P1: jsrec

GetRec Len P1: 51 codice objiter (ATTENZIONE: dimensione di o...

P2:

Nume:

Timeout: 60000

Descrizione:

*nume=1 => cancella tutto
nume<>1 => cancella solo iter
con iter.state='N'*

Mentre le altre chiamate sono identiche alle rispettive per il caso di eventi non periodici, con l'aggiunta del codice dell'iterazione nel rec nel campo **coditer**.

Se si rettifica la periodicità del piano è necessario innanzitutto eseguire una **undosendobjiterperiodic**

Gestione delle checklist (S5 - AUDIT)

Il flusso generale della gestione del suddetto object è il seguente:

1. newobjiter (tipo=P): crea il nuovo objiter e restituisce il **codice**
2. insert into project (codice,descrizio,codubi): inserisce in project con la descrizio ed il codubi (vedi sotto le query da svolgere)
3. initteam: inizializza il team, nel caso specifico individua "chi" dovrà eseguire la compilazione
4. generate: genera la checklist
5. answer: registra la risposta di ogni elemento della checklist
6. approve (da valutare se lo gestiremo): dichiara lo svolgimento eseguito dell'objiter

L'inserimento del progetto deve procedere svolgendo esattamente le seguenti query, sostituendo opportunamente il codice del progetto con il codice del objiter create (le tre query vanno fatte in un colpo solo):

```
INSERT INTO PROJECT (codice,descrizio,codubi) VALUES (8,'descrizione di
prova','0000000098'); -- (alloca il progetto)
INSERT INTO PRJMATGAP (CODPRJ,CODGRP,CODAREA,CODPROC) VALUES (7,1,1,1); --
(necessario per la gestione delle checklist)
INSERT INTO PRJCAT (codprj,codcat) SELECT 7,category.codice FROM category; --
(necessario per la gestione delle checklist)
```

Le tabelle coinvolte dai movimenti sono le seguenti:

- objiter: contiene la tipologia ed i riferimenti principali
- project: contiene alcune informazioni di dettaglio (descrizio, codubi, altro...)
- objteam: contiene il team (chi) che deve svolgere; il team avrà objteam.tipo=A
- iter: contiene le date di previsto svolgimento (tmsend) ed infine di svolgimento (tmfeed)
- prjtask: tabella intermedia dei task che collega l'istanza di checklist al progetto
- clista: contiene le checklist e le risposte correnti
- answer: contiene i commenti di ogni singola risposta e l'esito della risposta

Di seguito uno schema commentato della pagina degli Audit.

1.2.3.S3 - 5S - Audit sheet

5S Audit

associato ad un codubi con tipose='R'
il codubi va considerato dentro project

objiter.codoper
(operato.nominati)

objiter.datacre

last audit date!

category

generato
contando le
righe di
comment

(tramite clista)
checklist.comment

clista.curans

ultima answer della clista
answer.comment

Level	Item	Identifizieren & eliminieren was nicht benötigt wird	5S Level	Comment
1	1.1	1.1.1	1	
2	2.1	2.1.1	2	
3	3.1	3.1.1	3	
4	4.1	4.1.1	4	
5	5.1	5.1.1	5	
6	6.1	6.1.1	6	
7	7.1	7.1.1	7	
8	8.1	8.1.1	8	
9	9.1	9.1.1	9	
10	10.1	10.1.1	10	
11	11.1	11.1.1	11	
12	12.1	12.1.1	12	
13	13.1	13.1.1	13	
14	14.1	14.1.1	14	
15	15.1	15.1.1	15	
16	16.1	16.1.1	16	
17	17.1	17.1.1	17	
18	18.1	18.1.1	18	
19	19.1	19.1.1	19	
20	20.1	20.1.1	20	
21	21.1	21.1.1	21	
22	22.1	22.1.1	22	
23	23.1	23.1.1	23	
24	24.1	24.1.1	24	
25	25.1	25.1.1	25	
26	26.1	26.1.1	26	
27	27.1	27.1.1	27	
28	28.1	28.1.1	28	
29	29.1	29.1.1	29	
30	30.1	30.1.1	30	
31	31.1	31.1.1	31	
32	32.1	32.1.1	32	
33	33.1	33.1.1	33	
34	34.1	34.1.1	34	
35	35.1	35.1.1	35	
36	36.1	36.1.1	36	
37	37.1	37.1.1	37	
38	38.1	38.1.1	38	
39	39.1	39.1.1	39	
40	40.1	40.1.1	40	
41	41.1	41.1.1	41	
42	42.1	42.1.1	42	
43	43.1	43.1.1	43	
44	44.1	44.1.1	44	
45	45.1	45.1.1	45	
46	46.1	46.1.1	46	
47	47.1	47.1.1	47	
48	48.1	48.1.1	48	
49	49.1	49.1.1	49	
50	50.1	50.1.1	50	
51	51.1	51.1.1	51	
52	52.1	52.1.1	52	
53	53.1	53.1.1	53	
54	54.1	54.1.1	54	
55	55.1	55.1.1	55	
56	56.1	56.1.1	56	
57	57.1	57.1.1	57	
58	58.1	58.1.1	58	
59	59.1	59.1.1	59	
60	60.1	60.1.1	60	
61	61.1	61.1.1	61	
62	62.1	62.1.1	62	
63	63.1	63.1.1	63	
64	64.1	64.1.1	64	
65	65.1	65.1.1	65	
66	66.1	66.1.1	66	
67	67.1	67.1.1	67	
68	68.1	68.1.1	68	
69	69.1	69.1.1	69	
70	70.1	70.1.1	70	
71	71.1	71.1.1	71	
72	72.1	72.1.1	72	
73	73.1	73.1.1	73	
74	74.1	74.1.1	74	
75	75.1	75.1.1	75	
76	76.1	76.1.1	76	
77	77.1	77.1.1	77	
78	78.1	78.1.1	78	
79	79.1	79.1.1	79	
80	80.1	80.1.1	80	
81	81.1	81.1.1	81	
82	82.1	82.1.1	82	
83	83.1	83.1.1	83	
84	84.1	84.1.1	84	
85	85.1	85.1.1	85	
86	86.1	86.1.1	86	
87	87.1	87.1.1	87	
88	88.1	88.1.1	88	
89	89.1	89.1.1	89	
90	90.1	90.1.1	90	
91	91.1	91.1.1	91	
92	92.1	92.1.1	92	
93	93.1	93.1.1	93	
94	94.1	94.1.1	94	
95	95.1	95.1.1	95	
96	96.1	96.1.1	96	
97	97.1	97.1.1	97	
98	98.1	98.1.1	98	
99	99.1	99.1.1	99	
100	100.1	100.1.1	100	

Manual Value at the right section, the last set of values should be "coloured" at the left section values 1 – 5 (validation).

Total 5S Level on top is average of the lower section.

I passi 1, 2, 3, 6 sono svolti come indicato nel paragrafo precedente, mentre i passi 4, 5 come segue:

- generate

Opzioni agenti

Agente: prjman

Request: generate

P1: {"codice":12}

GetRec Req P1: newrec

GetRec Name P1: jsrec

GetRec Len P1: 51

P2: {"ciao":"ciao"}

GetRec Req P2: newrec

GetRec Name P2: jsrec

GetRec Len P2: 51

Nume:

Timeout: 60000

Descrizione:

- answer

Opzioni agenti

Agente: prjman

Request: answer

P1: {"codclista":5,"risp":4,"codoper":"admin","comment":"ok ok"}

GetRec Req P1: newrec

GetRec Name P1: jsrec

GetRec Len P1: 51

P2: {"ciao":"ciao"}

GetRec Req P2: newrec

GetRec Name P2: jsrec

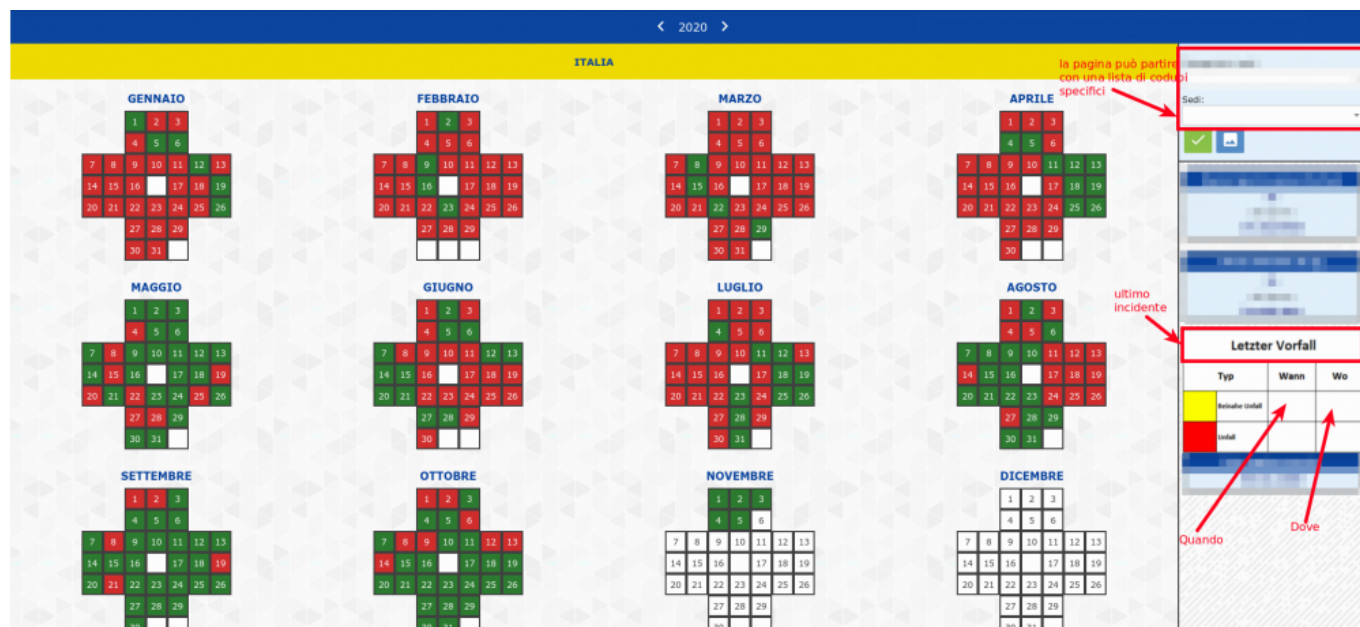
GetRec Len P2: 51

Nume:

Timeout: 60000

Descrizione:

Green Cross



Le Green Cross prevedono la visualizzazione di eventi di tipo:

- near miss (objiter.tipo='N'): colore giallo
- accident (objiter.tipo='A'): colore rosso

Oltre alla visualizzazione giornaliera dei dati viene anche indicata la data ed il luogo dell'ultimo incidente e dell'ultimo near miss (tabella a destra).

Al fine di avere un valore iniziale relativamente alla data ultimo incidente e data ultimo near miss, utilizziamo due campi aggiuntivi di ubicaz:

- ubicaz.tmnc: tm dell'ultimo incidente
- ubicaz.tnmn: tm ultimo near miss

```
ALTER TABLE ubicaz ADD COLUMN tmnc FLOAT, ADD COLUMN tnmn FLOAT;
comment ON COLUMN ubicaz.tmnc IS 'tm dell''ultimo incidente';
comment ON COLUMN ubicaz.tnmn IS 'tm dell''ultimo near miss';
```

L'AgenteDigCIC si occuperà di considerare questi valori, insieme a quelli derivanti dagli objiter creati, per fornire i dati alla GUI secondo gli schemi descritti di seguito.

La chiamata per ottenere i dati delle green cross è la seguente (greencross).

Test Agenti

Opzioni agenti

Agente	ciccic	Ask ▼
Request	greencross	
P1	{ "codubi": "0000000087,abt151", "year": 2020 }	
GetRec Req P1	newrec	
GetRec Name P1	jsrec	
GetRec Len P1	51	(ATTENZIONE: dimensione di ogni campo del rec,
P2		
Nume		
Timeout	60000	
Descrizione		

codubi divise da virgola

L'agente elabora le greencross di ogni location sottostante e restituisce la seguente lista.

Agents Diagnostic View

```
1 [{"datainc":20201020,"dayofmonth":20,"month":10,"dayofyear":301,"tipo":"A","ninc":2,"incdata":20201105,"incubi":"abt151","neardata":20201105,"nearubi":"abt151"},
{"datainc":20201029,"dayofmonth":29,"month":10,"dayofyear":302,"tipo":"N","ninc":1,"incdata":20201105,"incubi":"abt151","neardata":20201105,"nearubi":"abt151"},
{"datainc":20201105,"dayofmonth":5,"month":11,"dayofyear":309,"tipo":"A","ninc":2,"incdata":20201105,"incubi":"abt151","neardata":20201105,"nearubi":"abt151"},
{"datainc":20201105,"dayofmonth":5,"month":11,"dayofyear":309,"tipo":"N","ninc":1,"incdata":20201105,"incubi":"abt151","neardata":20201105,"nearubi":"abt151"}]
```

Data dell'incidente

tipologia: distinguiamo tra
N=nearmiss
A=accident
se nello stesso giorno ci sono
entrambi fa la cella rossa

numero di eventi

data ultimo incidente
e codubi ultimo
incidente

vengono scritti
su tutte le righe
ma il valore è
sempre lo stesso

data ultimo near miss
e codubi ultimo
near miss

CHIUDI

Quality

Action plan Quality

L'Action plan Quality ha la medesima impostazione degli Action plan definiti per la safety in particolare per la [Near Miss](#), [Accident](#), [long time S5](#) e [TPM, short time TPM](#). Il flusso è il medesimo definito per i tipi N,A,S,J; aggiungiamo un nuovo tipo che chiamiamo Q. Il resto del processo è esattamente lo stesso come pure la pagina cui si fa riferimento (vedi sotto).

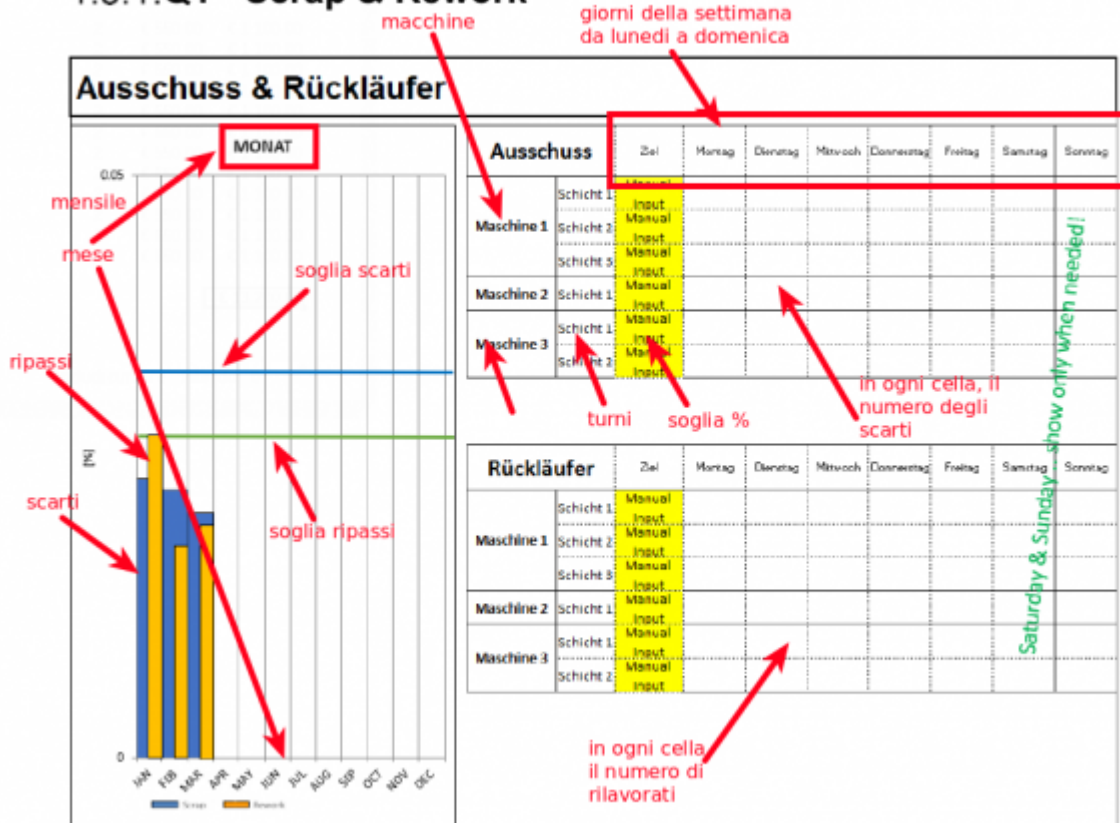
Aktions-Plan Qualität							
Nr.	Datum	Beschreibung	Hauptursache	Korrekturmaßnahmen	Wer	Datum	Erledigt
1							☒
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							

Grafici della qualità

Vengono definite 2 strutture basate su grafici per visualizzare i dati della qualità:

- 1. la prima visualizza, in un grafico a barre, la frazione di scarti/ripassi per ogni mese; inoltre, per ogni settimana, presenta una tabella, ove le colonne sono i giorni della settimana (lun, mar, ...) e le righe sono le macchine appartenenti al team; ogni cella contiene il numero di pezzi giornaliero di scarti/ripassi (2 tabelle)
- 2. la seconda visualizza il Pareto delle cause/numero di scarti/ripassi; in un unico grafico vengono affiancate sia le barre dei ripassi che quelle degli scarti con colore differente

1.3.1.Q1 - Scrap & Rework



Al fine di gestire le soglie di scarti/ripassi per location-turno (presenti nelle tabelle settimanali) viene creata la relazione **ubiturso** (soglie ubicaz-turnidet). Di seguito il file contenente la struttura e lo schema ER.

- ubiturso.sql



Al fine di inserire i dati nella ubiturso, l'utente potrà selezionare l'icona indicata sotto solo in corrispondenza di una specifica location di tipo linea di produzione.



Plant FBS DE Deutschland

Gestione Turni

Descrizione

Se selezionata linea di produzione apre una pagina dove l'operatore può inserire per ogni riga di turnidat la soglia scarti/ripassi (direttamente in tabella) inserendo i dati nella ubiturso

I dati relativi al report Q1 possono essere richiesti all'AgentDigCICSup attraverso la seguente chiamata:

Test Agenti

Opzioni agenti

Agente: Ask ▼

Request:

P1:

GetRec Req P1:

GetRec Name P1:

GetRec Len P1: (ATTENZIONE: dimensione di ogni car)

P2:

Nume:

Timeout:

Descrizione:

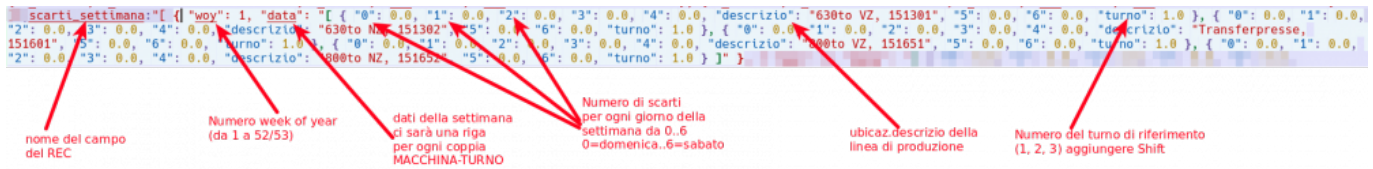
Ove in p1, troviamo il codice della location a partire dalla quale si desidera sviluppare il report e l'anno di riferimento. L'agente ritorna un REC contenente 3 campi che sono oggetti JSON:

- mensile: grafico dei dati mensili
- settimanale_scarti: tabella settimanale degli scarti contenente tutte le settimane dell'anno
- settimanale_ripassi: tabella settimanale dei ripassi contenente tutte le settimane dell'anno

Di seguito il JSON di "mensile".



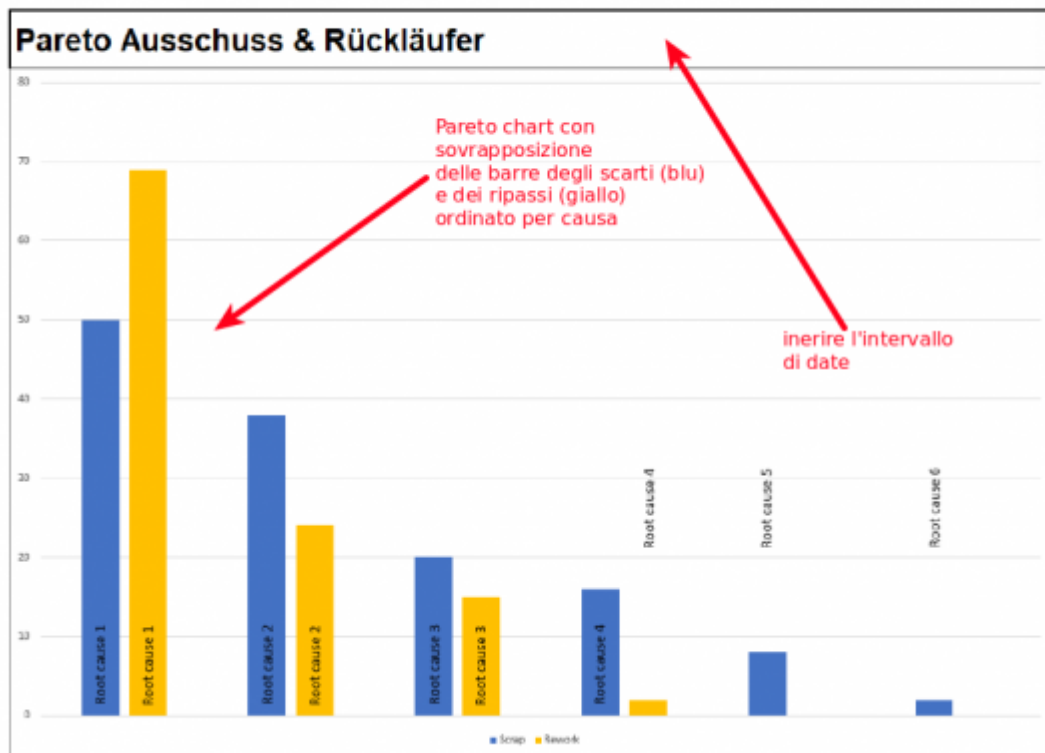
Di seguito il JSON di "settimanale_scarti" (il JSON di settimanale_ripassi è uguale).



@ **Fix Me!** Nel JSON Settimanale è stato integrato un campo aggiuntivo per **ubicazione-turno** che si chiama "soglia"; tale campo rappresenta la soglia giornaliera in termini di numero di scarti o ripassi (ubiturso.nsca, ubiturso.nrip).

Q2

1.3.2.Q2 – Pareto Scrap / Rework



E' possibile estrarre la statistica Q2 tramite l'AgentDigCICSup con la seguente richiesta, riferita ad una specifica codubi e ad un intervallo temporale.

Test Agenti

Opzioni agenti

Agente: Ask ▾

Request:

P1: esempio newrec: {a:10,b:'ciao'}

GetRec Req P1:

GetRec Name P1:

GetRec Len P1: (ATTENZIONE: dimensione di ogni campo del rec, non di tutto il P1)

P2:

☐ Contiene Caratteri Speciali

Nome:

Timeout:

Descrizione:

La risposta è un rec contenente per ogni causale identificata di scarto il numero di scarti (scarti) ed il numero di ripassi (ripassi), oltre al codice e descrizione della causale di scarto/ripasso.

```
1 [{"descrizio":"Reisser",caudef:"crack",scarti:518,ripassi:0},{descrizio:"Falten",caudef:"fold",scarti:44,ripassi:0},{descrizio:"Material",caudef:"material",scarti:43,ripassi:0},{descrizio:"Werkzeug",caudef:"tool",scarti:79,ripassi:0}]
```

Quality Alarms

Di seguito lo schema da realizzare per i quality alarms.

1.3.3.Q3 - Quality incidents

Qualitäts-Alarm

Prozess Beschreibung: Abkant-Pressa 230ts

Kunde (Intern/Extern): Pulver Beschichtung

Entdecktes Problem / Hauptgrund: Clinching quality varies in top right corner/Inbalance in frame

desrizione del processo (macchina) *problema riscontrato* *causa principale del problema*

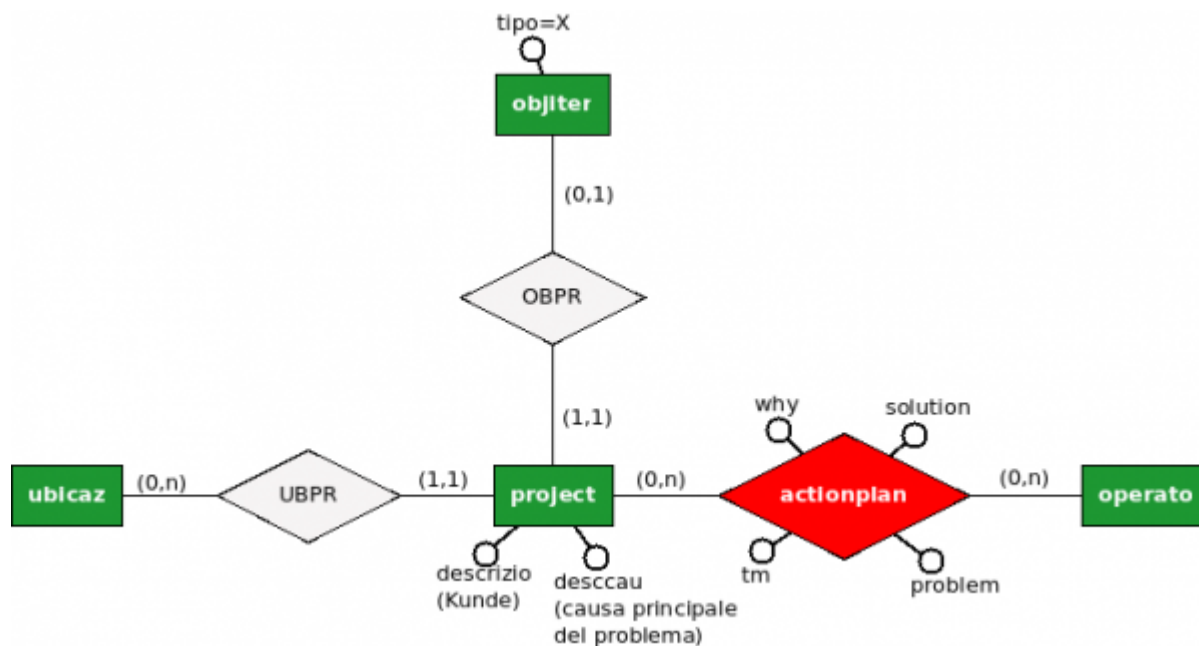
soluzione *perchè*

Nr.	Problem	Lösung	Warum?	Wer	Erledigt
1	Imbalance in frame for clinching	Rebuilt stable adjustable (for each corner) fra	Prevent clinching problems	CW	30.04.2016
2					
3					
4					
5					

quando *chi*

Like a library, all warnings displayed with an interval of ca. 15 seconds. Editing by team leader at his pc.

Verrà inserito come objiter di tipo X, e viene allocata una nuova tabella relazionata alla project secondo il seguente schema.



L'oggetto non sarà soggetto ad iterazioni e non richiede la costruzione del team. Il seguente file crea la actionplan:

- actionplan.sql

Costs

Il modulo Costs presenta le seguenti funzioni:

- C1 - OEE
- C2 - STOPPAGE
- C3 - CHANGEOVER
- C4 - ACTION PLAN

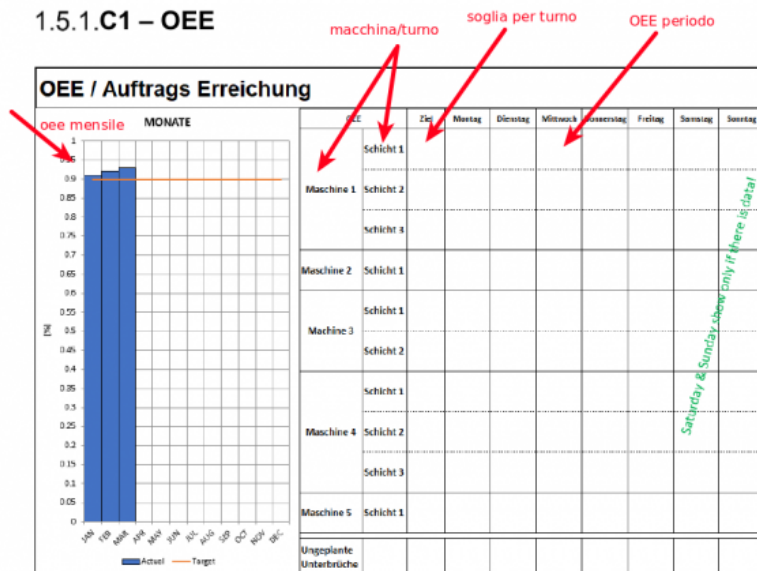
Il modulo C4 - Action Plan sarà gestito come gli altri action plan con tipo='O'.

C1 - OEE

Al fine di realizzare il grafico C1 sarà necessario attivare per lo specifico plant il calcolo dell'OEE per turno per ogni location del plant. In particolare si dovrà tenere in conto la seguente raccomandazione da verificare prima dell'avviamento:

- **LE LINEE DEL PLANT SONO SOGGETTE A CAMBIO DEL TURNO:** in tal caso si dovrà prevedere, al momento del caricamento del plant, la possibilità di modificare il turno in OEECONF per le linee (da sviluppare solo durante l'eventual estensione)

1.5.1.C1 – OEE



Data from NeXT OEE. Number of lines related to machines and shifts used.

Saturday & Sunday only displayed when used.

Target to be set by team leader depending on line mix situation.

Il grafico sopra ha il medesimo schema del grafico Q1 della qualità. A sx abbiamo l'andamento mensile totale, mentre a dx abbiamo calendari settimanali per macchina/turno ove sono indicati i valori dell'OEE e le relative soglie.

I dati sono forniti allo stesso modo dall'AgentDigCICSup attraverso la seguente chiamata.

Test Agenti

Opzioni agenti

Agente: ciccic Ask

Request: C1

P1: {"codubi":"abt110","year":2020} esempio newrec:

GetRec Req P1: newrec

GetRec Name P1: jsrec

GetRec Len P1: 51 (ATTENZIONE: dimensione di ogni campo del rec, non di tutto il .)

P2: Contiene C

Nume: location

Timeout: 60000

Descrizione:

L'agente risponde con un rec contenente i due campi seguenti:

- mensile: array JSON con la lista dei mesi dell'anno e l'OEE + soglia di ogni mese

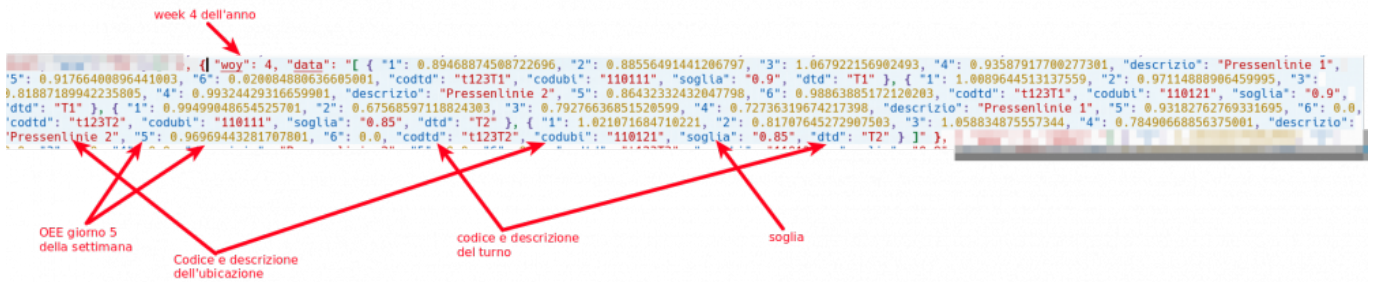
```
[{"mensile": [{"month": 1, "oee": 0.87929738681913794, "soglia": 0.0}, {"month": 2, "oee": 0.85855621667673432, "soglia": 0.0}, {"month": 3, "oee": 0.7142867977347751, "soglia": 0.0}, {"month": 4, "oee": 0.67854398079730959, "soglia": 0.0}, {"month": 5, "oee": 0.77728245864154566, "soglia": 0.0}, {"month": 6, "oee": 0.78879202601415154, "soglia": 0.0}, {"month": 7, "oee": 0.82899822375269993, "soglia": 0.0}, {"month": 8, "oee": 0.78613756787350142, "soglia": 0.0}, {"month": 9, "oee": 0.79749551998945567, "soglia": 0.0}, {"month": 10, "oee": 0.83796592482137877, "soglia": 0.0}, {"month": 11, "oee": 0.75929513614108179, "soglia": 0.0}]}]
```

numero del mese

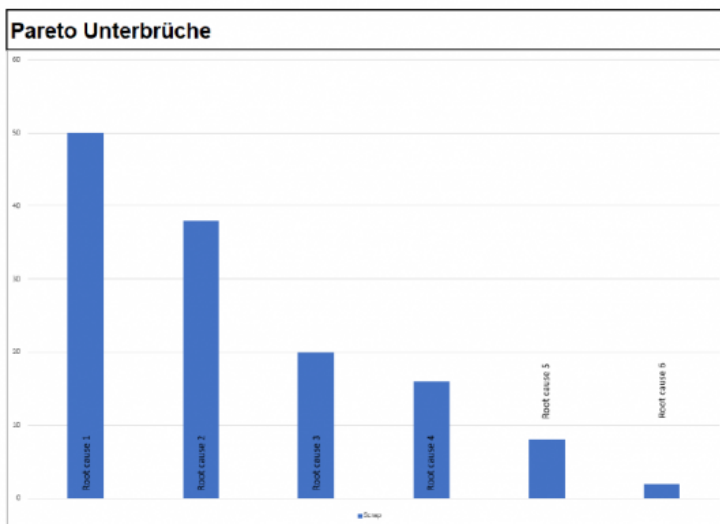
valore dell'oee

soglia

- oee_settimana: array JSON contenente le settimane dell'anno (woy=1..53) ed un campo data
 - il campo data a sua volta contiene un array JSON come segue



C2 - STOPPAGES



All Stoppages during the actual week for all machines associated to the "Team board".

Si tratta di un semplice Pareto delle fermate accessibile da qualsiasi livello della gerarchia. Per ottenere i dati è sufficiente inoltrare all'AgentDigCICSup, la seguente chiamata.

Test Agenti

Opzioni agenti

Agente: Ask ▼

Request:

P1:

GetRec Req P1:

GetRec Name P1:

GetRec Len P1: (ATTENZIONE: dimensione di ogni carr)

P2:

Nume:

Timeout:

Descrizione:

location

intervallo di date

L'agente risponde con la seguente semplice struttura (rec).

```

[{"f":296454,"caufm":"Presswerk","descrizio":"Maschine&Werkzeug justieren / Machine&Tool adjustments"}, {"f":244465,"caufm":"fermo","descrizio":"UNBEKANNTER Stillstand"}, {"f":229960,"caufm":"U003","descrizio":"Material/Qualität"}, {"f":212818,"caufm":"PW1 Raziol","descrizio":"Paketwechsel Raziol"}, {"f":161064,"caufm":"PW1 000","descrizio":"0_Raziol INV1105"}, {"f":129315,"caufm":"PW2 000","descrizio":"0_Raziol INV1979"}, {"f":108080,"caufm":"U002PW1002M","descrizio":"Maschine"}, {"f":79734,"caufm":"U002PW2002M","descrizio":"Maschine"}, {"f":75191,"caufm":"U002PW1001M","descrizio":"Maschine"}, {"f":74696,"caufm":"U002PW2001M","descrizio":"Maschine"}, {"f":68810,"caufm":"U005","descrizio":"Schichtübergabe"}, {"f":51086,"caufm":"U002PW2004M","descrizio":"Maschine"}, {"f":47079,"caufm":"U002PW1002T","descrizio":"Werkzeug"}, {"f":46208,"caufm":"PW1 006","descrizio":"R2_Roboter INV3628"}, {"f":46044,"caufm":"PW2 a100","descrizio":"R3V_Roboter INV1004"}, {"f":45058,"caufm":"PW1 005","descrizio":"R1_Roboter INV3629"}, {"f":35223,"caufm":"U002PW1004M","descrizio":"Maschine"}, {"f":34378,"caufm":"U002PW2003T","descrizio":"Werkzeug"}, {"f":29417,"caufm":"U002PW1003T","descrizio":"Werkzeug"}, {"f":27060,"caufm":"PW2 a120","descrizio":"Überlaufstanze"}, {"f":25695,"caufm":"U002PW2003M","descrizio":"Maschine"}, {"f":24537,"caufm":"U002PW1004T","descrizio":"Werkzeug"}, {"f":23111,"caufm":"PW2 070","descrizio":"R2H_Roboter INV1093"}, {"f":19751,"caufm":"PW1 007","descrizio":"RE_Einlegeroboter INV3632"}, {"f":17023,"caufm":"U002PW1001T","descrizio":"Werkzeug"}, {"f":17004,"caufm":"PW2 080","descrizio":"R2V_Roboter INV1084"}, {"f":14491,"caufm":"U002PW1003M","descrizio":"Maschine"}, {"f":13748,"caufm":"U002PW2004T","descrizio":"Werkzeug"}, {"f":13148,"caufm":"PW2 090","descrizio":"R3H_Roboter INV1095"}, {"f":8321,"caufm":"U002PW2001T","descrizio":"Werkzeug"}, {"f":7845,"caufm":"PW2 050","descrizio":"R1V_Roboter INV3633"}, {"f":7115,"caufm":"U002PW2002T","descrizio":"Werkzeug"}, {"f":7063,"caufm":"PW2 060","descrizio":"R1H_Roboter INV1079"}, {"f":4126,"caufm":"PW2 a110","descrizio":"R4V_Roboter INV1098"}]

```

tempo di fermo in secondi causa del fermo

C3 - Changeover

La pagina del Changover dovrà presentare un blocco dati per ogni linea appartenente ad un determinato Team; in un pagina si dovranno presentare massimo 4 linee alla volta, con la possibilità di poter scorrere manualmente oppure automaticamente.

I dati saranno prodotti dall'AgentDigCICSup, in base al codice del Team (codubi) ed ad un intervallo temporale (tmin, tmfi). L'agente produce un rec contenente tutti i dati ordinati per codubi delle linee di produzione e per tm.

Di seguito uno schema dell'interfaccia.



I dati presenti in ogni blocchetto macchina sono i seguenti:

- nome della linea/macchina
- numero di eventi di changeover (numero di righe del rec ritornato riferiti alla linea)
- quantità media ordini del periodo (presente nella colonna qtameop del rec dell'agente)
- durata totale changover (somma dei tmdiff presenti nel rec per la linea)

- durata media changeover (media dei tmdiff presenti nel rec per la linea)
- mediana della durata di changeover (mediana dei tmdiff presenti nel rec per la linea)
- grafico
 - serie 1: x = tm del rec per la linea, y = tmdiff del rec per la linea
 - serie 2: y = media
 - serie 3: y = mediana
 - dati aggiuntivi: data inizio, fine, codice e descrizione dell'articolo in lavorazione (questi ultimi si trovano nel rec per la linea, come JSON nel campo "catego" della riga corrispondente)

Di seguito la richiesta da inviare all'AgentDigCICSup.

Test Agenti

Opzioni agenti

Agente

ciccic

Ask ▼

Request

C3

P1

{"codubi":"abt110","tmin":1577833200,"tmfi":1605394800}

GetRec Req P1

newrec

GetRec Name P1

jsrec

GetRec Len P1

51

(ATTENZIONE: dimensione di ogni campo del rec,

P2

Nume

Timeout

60000

Descrizione

La risposta dell'agente è la seguente.

```
1 [{"codice":8868231,"tm":1578404160,"tmdiff":4,"caufe":"wechsel","dcau":"Werkzeug wechseln / Tool change","catego":[{"codcate":"\\101.0288.025\\","dcate":"\\ESN 614 1 1\\2\\2\\\\" NWK REV RV Nova\\"}]","codubi":"1111","dubi":"Pressenlinie","qtameop":721,"001859"},{"codice":8865293,"tm":1578485600,"tmdiff":295,"caufe":"wechsel","dcau":"Werkzeug wechseln / Tool change","catego":[{"codcate":"\\101.0389.698\\","dcate":"\\ESN 614 1 1\\2\\2\\\\" NWK REV RV\\"}]","codubi":"1011","dubi":"Pressenlinie 1","qtameop":721,"001859"}],
```

Rec ordinato per codubi e tm

tm e tmdiff dell'evento

codice e descrizione della causale

JSON contenente uno o più articoli eventualmente presenti nel periodo (codcate, dcate)

codice e descrizione della ubicaz

Quantità media ordini di produzione ripetuto su tutte le righe della linea (sempre stesso valore)

Delivery

Il modulo Delivery presenta le seguenti funzioni:

- D1 - Order achievement
- D2 - Pareto Order achievement / losses
- D3 - ???
- D4 - ACTION PLAN

Il modulo D4 - Action Plan sarà gestito come gli altri action plan con tipo='D'.

D1 - Order Achievement

La logica di funzionamento di questa tabella è la medesima della tabella dell'OEE in C1; l'unica differenza è che le soglie sono soglie giornaliere e per macchina/turno. Questo sarà gestito nel rec ritornato dall'agente.

Auftrags Erreichung		settimana													
		Montag		Dienstag		Mittwoch		Donnerstag		Freitag		Samstag		Sonntag	
		Ziel	Erreicht	Ziel	Erreicht	Ziel	Erreicht	Ziel	Erreicht	Ziel	Erreicht	Ziel	Erreicht	Ziel	Erreicht
Maschine 1	Schicht 1	450	460	440	435										
	Schicht 2	340	315	350	335										
	Schicht 3	430	430	440	435										
Maschine 2	Schicht 1	5500	5356	5400	5700										
Maschine 3	Schicht 1	430	437	440	435										
	Schicht 2														
Maschine 4	Schicht 1														
	Schicht 2	325	335	330	325										
Verpasste Stücke			-142		265										

La chiamata all'agente restituisce direttamente un rec con la coppia:

- woy: numero della settimana dell'anno (per tutto l'anno)
- data: JSON contenente i dati della settimana

La richiesta all'AgentDigCICSup è la seguente.

Test Agenti

Opzioni agenti

Agente
ciccic
Ask

Request
D1

P1
{"codubi":"abt110","year":2020}
esempi

GetRec Req P1
newrec

GetRec Name P1
jsrec

GetRec Len P1
51
(ATTENZIONE: dimensione di ogni campo del rec, non c

P2
☐ C

Nume

Timeout
60000

Descrizione

La risposta dell'agente ha la medesima struttura della parte "settimanale" del report C1. Da notare che in ogni giorno della settimana (0, 1, ..., 6), questa volta troviamo due numeri interi divisi da una virgola:

- il primo numero intero è il numero di pezzi prodotto nel turno
- il secondo numero intero è il target del turno in base al takt time delle lavorazioni effettuate

```
[{"woy":1,"data":{"woy":2,"data":{"2":{"78,75","178,1426","4":{"1295,1385","descrizio":"Pressentintie 1","5":{"1340,1373","6":{"1125,1032","codtd":"1123T1","codubi":"110111","soglia":"0.0","tmf":"1116,1162","3":{"1151,1241","4":{"1196,1303","descrizio":"Pressentintie 1","5":{"1356,1325","6":{"1123T2","codubi":"111111","soglia":"0.0","dtd":"T2"},"...]}},...]}},...]}},...]}},...]}],...}]
```

dati di martedì
(giorno 2)

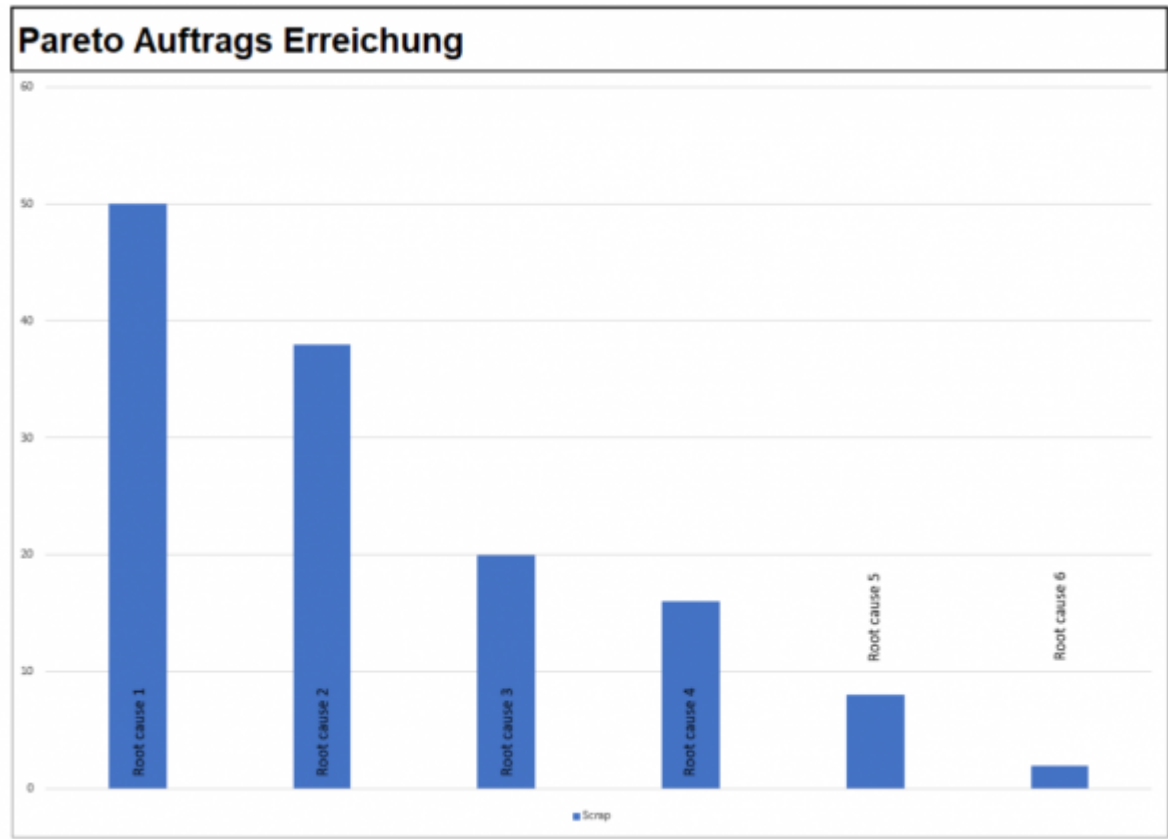
numero di pezzi

soglia

mercoledì ...

D2 - Pareto Order achievement / losses

Semplice pareto delle cause di speed-losses, riferito ad uno specifico Team (codubi in generale) ed ad uno specifico intervallo temporale (tmin, tmfi).



A tale scopo si può chiedere all'AgentDigCICSup la seguente ask.

Test Agenti

Opzioni agenti

Agente

ciccic

Ask

Request

D2

P1

{"codubi":"abt151","tmin":1577833200,"tmfi":1605394800}

esempi

GetRec Req P1

newrec

GetRec Name P1

jsrec

GetRec Len P1

51

(ATTENZIONE: dimensione di ogni campo del rec, non

P2

Nume

Timeout

60000

Descrizione

A cui l'agente risponde fornendo la lista delle cause di speedlosses riscontrate nel periodo (INRA) con la somma del numero di pezzi persi, l'elenco è già ordinato in modo decrescente di numero di pezzi persi.

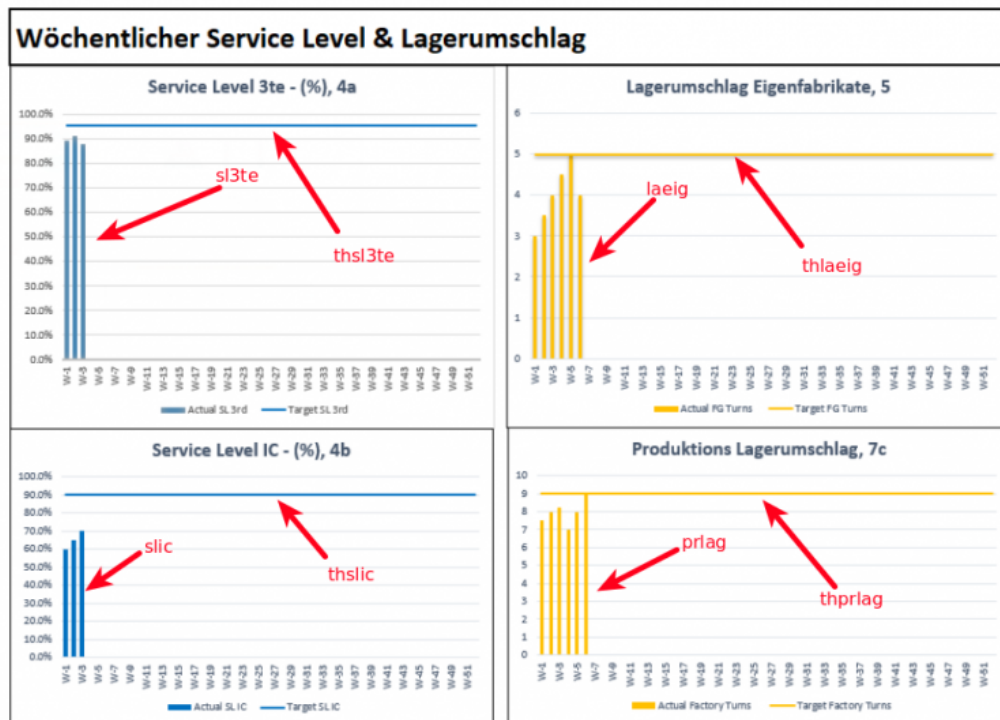
```
1 [{"q:315,caufm:"S01895",descrizio:"Materialtest"},{"q:47,caufm:"S01803",descrizio:"Daten aus ERP nicht aktuell"},{"q:19,caufm:"S01801",descrizio:"Schlechtes Material (Ziehfähigkeit)"}, {"q:18,caufm:"S01802",descrizio:"Temporärer/Untrainierter Mitarbeiter"}, {"q:7,caufm:"S01804",descrizio:"0-Serie/1st-Serie"}]
```

Numero Causa

D4 - SERVICE LEVEL

3.4.4.D4 – Stock turn & Service Level

dimension-servicelevel d1=codubi (codice plant) d2="servicelevel" year=anno, raggruppato per woy



i campi:
thsl3te, thlaeig, thslic, thprlag
vanno inseriti nella ubiturso
e inseriti in modo da poter
essere configurati tramite
il configuratore

I dati delle soglie non vengono
salvati nel telecontrol quindi
la GUI li recupera mettendosi
in join con la ubiturso e d1 di
dimension

Aggiungere in **ubicaz** i campi:

- thmonslev, thdayslev con relativi commenti

Gestire l'inserimento di queste soglie ed anche delle soglie thmonscrap e thmonrip, in una pagina di configurazione associata al plant.

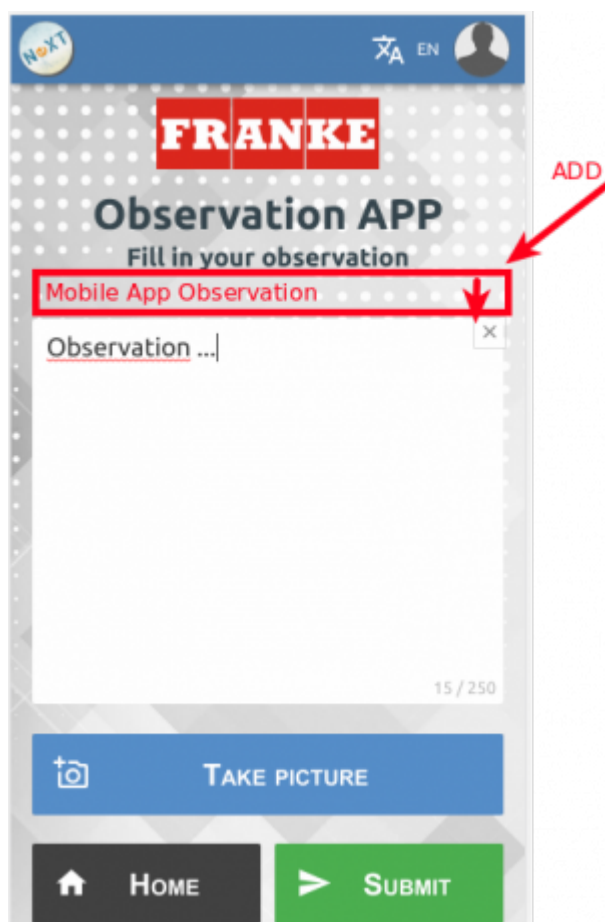
```
ALTER TABLE ubicaz ADD COLUMN thmonslev DOUBLE PRECISION;
ALTER TABLE ubicaz ADD COLUMN thdayslev DOUBLE PRECISION;
comment ON COLUMN ubicaz.thmonslev IS 'Soglia mensile Service Level';
comment ON COLUMN ubicaz.thdayslev IS 'Soglia giornaliera Service Level';
```

Mobile App

Di seguito la pagina di accesso all'APP.



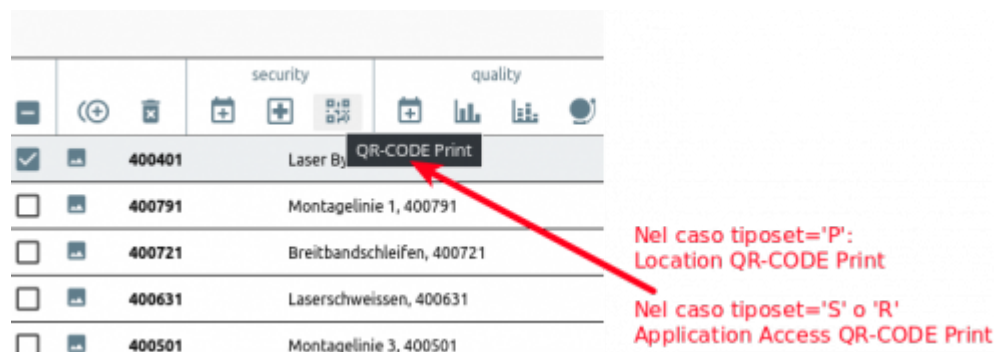
Dopo aver selezionato la location si accede alla pagina di registrazione dell'osservazione.



Aggiungere la scelta del tipo di evento per default "Mobile app observation".

Accesso all'App

Si prevede un nuovo QR-CODE che sarà possibile stampare direttamente dall'applicazione e che contiene il link per l'accesso. Questo QR-CODE potrà essere stampato dalla seguente funzione accessibile per tiposet='S' o tiposet='R'.

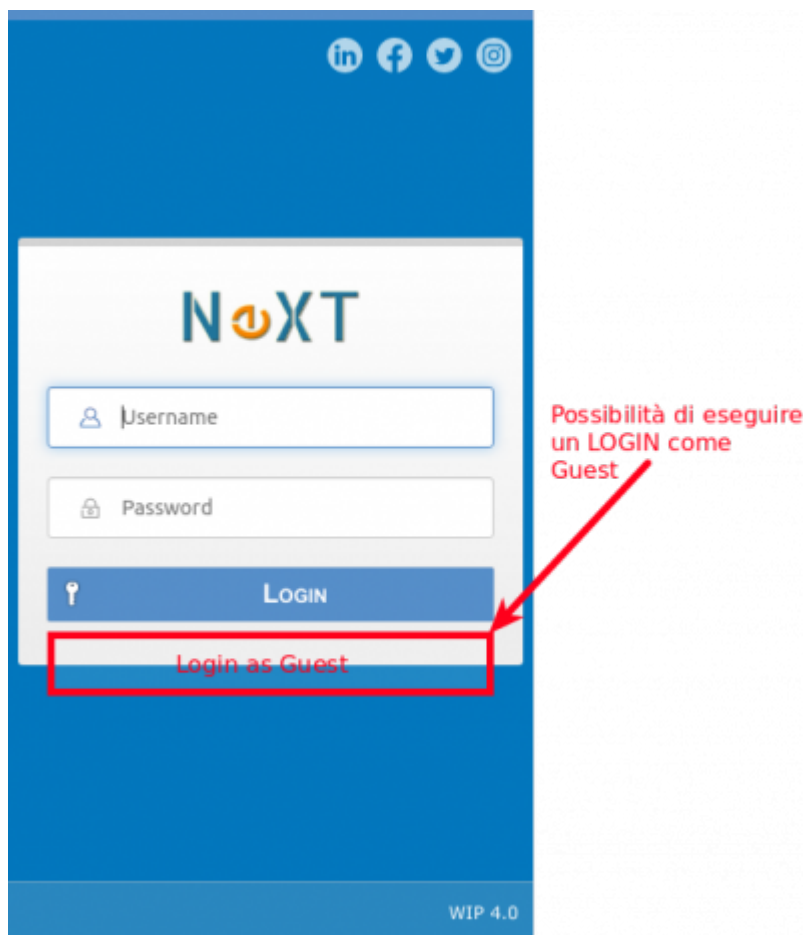


L'accesso può essere svolto attraverso una delle seguenti modalità:

- per ogni QR-CODE della location, si aggiunge sulla stampa un piccolo QR-CODE per accedere all'App (contiene il link)
- in varie posizioni nel plant è presente il QR-CODE per l'accesso all'applicazione indipendentemente dal QR-CODE della zona
- scaricare da Google Play store l'apk solo per android ed usare l'apk
- digitare manualmente nel browser il link di accesso

Login all'app

Una volta acceduto all'App essa richiede sempre il login. Attraverso un apposito parametro di accesso l'applicazione può permettere il login come Guest; in questo modo anche utenti esterni e non registrati potranno accedere all'applicazione e registrare osservazioni.



Scelta manuale della location

Condizione 1 - Guest

Il Guest entra in una sessione di scelta che poi lo porterà nella pagina di compilazione esattamente come fa la lettura del qr-code.

La pagina di scelta e' fatta come segue:

- ha una riga di ricerca (lente)
- sotto ha una lista di location

L'applicazione svolge la seguente sequenza di operazioni:

1. al primo ingresso le location sono i plant (tiposet='S')
2. se clicco su un plant entro nel livello sotto team (tiposet='R')
3. se clicco sul team entro nel livello linea (tiposet='P')
4. se clicco sulla linea entro in compilazione

Condizione 2 - Operatore loggato

Tutto come condizione 1, ma l'operatore loggato entra nella lista dei plant visualizzando solo i plant a lui associati, se in workpla ha piu' di un plant, mentre se ha un solo plant associato allora entra direttamente nei teams. Se non ha alcun plant associato allora parte da tutti i plant (o non vede nulla ?).

3 tipi di Observation

Aggiungere i seguenti campi.

```
ALTER TABLE project ADD COLUMN safetype VARCHAR(2), ADD COLUMN whathappen
VARCHAR(1024), ADD COLUMN potinj VARCHAR(1024), ADD COLUMN gravity VARCHAR(2), ADD
COLUMN pertype VARCHAR(1024);
comment ON COLUMN project.safetype IS 'relativo a objiter.tipo='N'', indica:
N=near miss,A=unsafe action,C=unsafe condition,E=unsafe equipement';
comment ON COLUMN project.whathappen IS 'relativo a objiter.tipo='N'', indica:
what happen';
comment ON COLUMN project.safetype IS 'relativo a objiter.tipo='N'', indica:
potenziali incidenti';
comment ON COLUMN project.gravity IS 'relativo a objiter.tipo='N'', indica:
R=red,Y=yellow,G=green';
comment ON COLUMN project.pertype IS 'relativo a objiter.tipo='5S' oppure
'TPM', indica: Performance Type';
```

Aggiungere inoltre la seguente tabella da utilizzare per la pagina di safety.

```
CREATE TABLE public.prjunsafe (
    codprj INTEGER NOT NULL,
    pos INTEGER NOT NULL,
    descrizio CHARACTER VARYING(1024)
);
COMMENT ON TABLE public.prjunsafe IS 'How can we prevent the occurrence/unsafe
situation?';
COMMENT ON COLUMN public.prjunsafe.codprj IS 'codice progetto';
COMMENT ON COLUMN public.prjunsafe.pos IS 'sequenza per un dato progetto';
ALTER TABLE ONLY public.prjunsafe
    ADD CONSTRAINT ix_prjunsafe_key PRIMARY KEY (codprj, pos);
ALTER TABLE ONLY public.prjunsafe
    ADD CONSTRAINT ix_prjunsafe_codprj FOREIGN KEY (codprj) REFERENCES
public.project(codice) ON UPDATE CASCADE ON DELETE CASCADE;
```

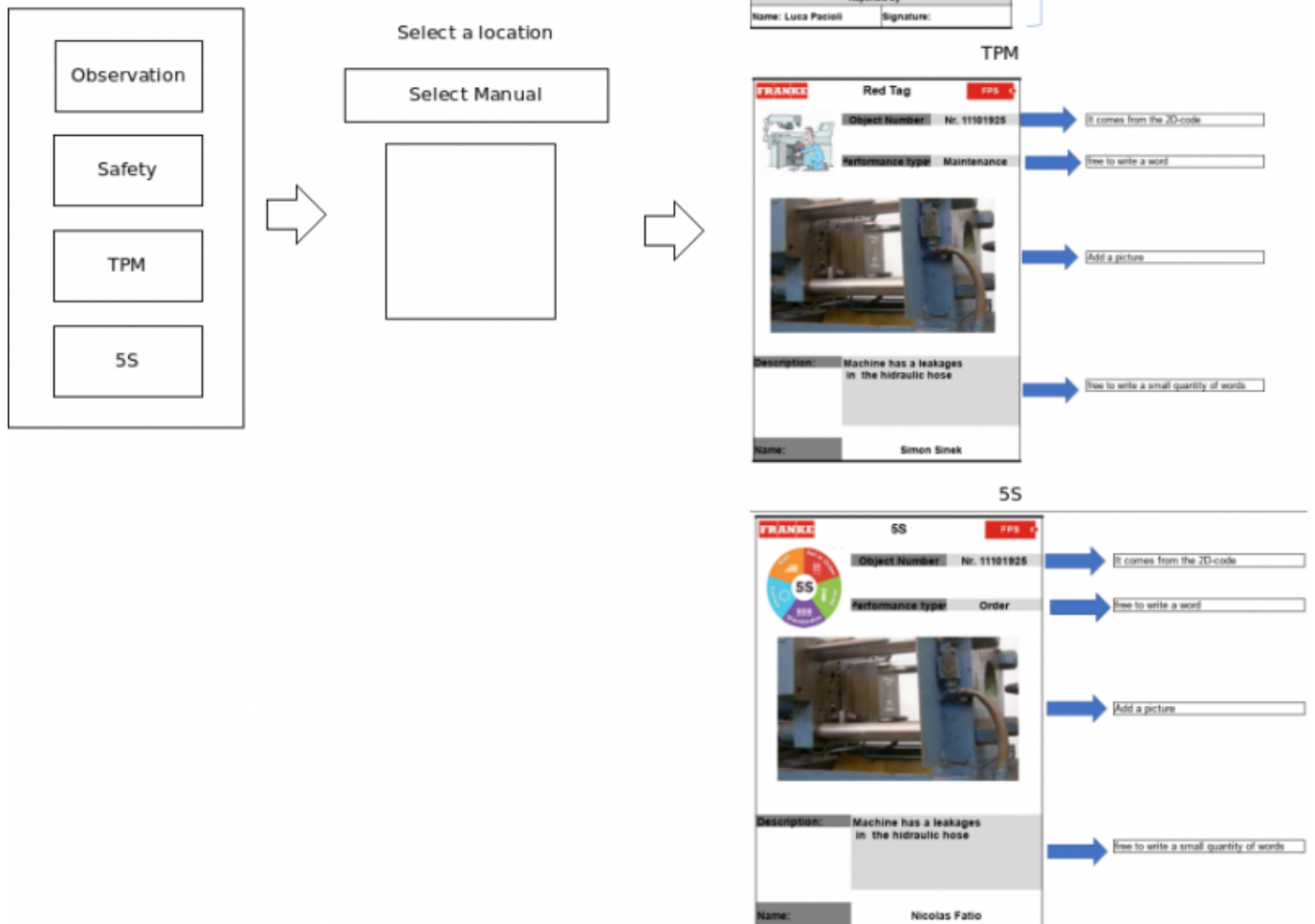
Aggiungere alle anags la tipologia di unsafe

```
INSERT INTO public.anags VALUES ('it', 'unsafe-type', 'A', 'Unsafe Action', 1, '',
'');
INSERT INTO public.anags VALUES ('it', 'unsafe-type', 'C', 'Unsafe Condition', 2,
'', '');
INSERT INTO public.anags VALUES ('it', 'unsafe-type', 'E', 'Unsafe Equipment', 3,
'', '');
INSERT INTO public.anags VALUES ('it', 'unsafe-type', 'N', 'Near Miss', 4, '', '');
INSERT INTO public.anags VALUES ('en', 'unsafe-type', 'A', 'Unsafe Action', 1, '',
'');
INSERT INTO public.anags VALUES ('en', 'unsafe-type', 'C', 'Unsafe Condition', 2,
'', '');
INSERT INTO public.anags VALUES ('en', 'unsafe-type', 'E', 'Unsafe Equipment', 3,
'', '');
INSERT INTO public.anags VALUES ('en', 'unsafe-type', 'N', 'Near Miss', 4, '', '');
INSERT INTO public.anags VALUES ('de', 'unsafe-type', 'A', 'Unsafe Action', 1, '',
'');
```

```
INSERT INTO public.anags VALUES ('de', 'unsafe-type', 'C', 'Unsafe Condition', 2,
'', '');
INSERT INTO public.anags VALUES ('de', 'unsafe-type', 'E', 'Unsafe Equipment', 3,
'', '');
INSERT INTO public.anags VALUES ('de', 'unsafe-type', 'N', 'Near Miss', 4, '', '');
INSERT INTO public.anags VALUES ('pl', 'unsafe-type', 'A', 'Unsafe Action', 1, '',
'');
INSERT INTO public.anags VALUES ('pl', 'unsafe-type', 'C', 'Unsafe Condition', 2,
'', '');
INSERT INTO public.anags VALUES ('pl', 'unsafe-type', 'E', 'Unsafe Equipment', 3,
'', '');
INSERT INTO public.anags VALUES ('pl', 'unsafe-type', 'N', 'Near Miss', 4, '', '');
```

L'app permette di selezionare una tra le seguenti funzioni:

- Observation: apre la pagina delle osservazioni già realizzata
- Safety: apre la pagina safety vedi sotto
- TPM: apre la pagina TPM vedi sotto
- 5S: apre la pagina 5S vedi sotto



CIC

Safety

S1 - Safety Cross

E' necessario integrare il **record senza infortuni** e la data da cui comincia il record, a tal fine aggiungiamo la possibilità di poter inserire nell'anagrafica ubicaz i seguenti campi:

```
ALTER TABLE ubicaz ADD COLUMN hrecnr INTEGER, ADD COLUMN hrecgg CHARACTER VARYING(9);
comment ON COLUMN ubicaz.hrecnr IS 'Record in termini di numero di giorni senza
```

```
infortuni per il Team tipsoet='R';
comment ON COLUMN ubicaz.hrecgg IS 'Data di inizio del record senza infortuni';
```

L'agente restituirà per ogni green-cross due campi aggiuntivi:

- hrecnr: numero di giorni senza infortuni
- hrecgg: data di inizio del record

Questi dati dovranno essere visualizzati a destra come segue.



Un unico BOX riferito all'oggetto (codubi) sul quale stiamo lavorando.

L'AgentDigCICSup ritorna i valori necessari, vedi sotto.

E' necessario configurare i dati necessari al calcolo del record infortuni di seguito.

tmnc	double precision	0	Visualizza	Modifica	Privilegi	Elimina	Istante di tempo dell'ultimo incidente registrato accaduto nella location
tmnm	double precision	0	Visualizza	Modifica	Privilegi	Elimina	Istante di tempo dell'ultimo near miss accaduto nella location
hrecnr	integer		Visualizza	Modifica	Privilegi	Elimina	Record in termini di numero di giorni senza infortuni per il Team tipsoet='R'
hrecgg	character varying(9)		Visualizza	Modifica	Privilegi	Elimina	Data di inizio del record senza infortuni

Assegnare questi dati solo ai team tramite una apposita scheda accessibile dal lato dx della pagina

Aggiungere a tale scopo l'icona di accesso alla apposita scheda come segue.



L'ulteriore campo visibile è la data di avvio della location necessario per il calcolo dei record; questo dovrà essere associato all'elemento selezionato e dovrà essere editabile, una volta modificato va su ubicaz.datareg.

S3 - TPM / 5S Assessment (Kami Shabai)

Si tratta di pianificazione kami shabai (modello da lean manufacturing). In pratica si pianificano una serie di attività che hanno certi obiettivi definiti dalla carta delle attività. Le attività vengono assegnate a un team e vengono pianificate nel tempo con una certa frequenza (settimanali, giornaliere, altro...).

Il tipo di objiter dedicato ai kami shabai è **K** ([K]ami shabai).

Il team viene inserito tra gli **owner**, il **team leader** viene inserito come objiter.codoper ed anche nel objteam con tipo='O'; gli altri partecipanti al team vanno inseriti solo in objteam.

A partire da un calendario mensile è possibile vedere giornalmente la pianificazione di queste attività. Un auditor ha il compito di selezionare una cella in cui è presente una attività e visualizzare la T-CARD. La T-CARD è una carta che descrive alcune informazioni dell'attività. L'auditor ha il compito di ispezionare l'area interessata dall'attività e vedere se trova proprietà corrispondenti a quelle previste dalla T-CARD; se c'è corrispondenza allora da un feedback positivo (MEET=1) e la card si colora di verde, altrimenti da un feedback negativo (MEET=0) e la card si colora di rosso. Cliccando sulla cella è possibile visionare tutte le T-CARD colorate che sono già state assegnate a tale area con il rispettivo esito.

Quando l'auditor clicca su una cella, viene automaticamente inserito nel **team di approver** per l'objiter interessato, se non è già presente; in questo modo l'utente dall'auditor potrà approvare/rigettare la specifica iter.

I tipi di attività che si vedono dal calendario dell'esempio sono:

- QA
- C
- FPS
- Safety

Quando si alloca una nuova card di un certo tipo, si dovrà applicare un contatore di card di quel tipo e incrementarlo. A tale scopo la CARD va creata direttamente tramite AgentDigCICSup; l'agente riceve in ingresso i dati della **newobjiter** piu' il tipo di card ed avrà il compito di svolgere tutte le operazioni di inizializzazione dell'objiter e del project con l'assegnazione dell'ID card. Di seguito la modalità di comunicazione.

Opzioni agenti	
Agente	ciccic Ask ▼
Request	newcard
P1	{"tipo":"K","codoper":"admin","tipocar":"QA",codubi:"FKT"} es
GetRec Req P1	newrec
GetRec Name P1	jsrec
GetRec Len P1	51 (ATTENZIONE: dimensione di ogni campo del rec, r
P2	[
Nume	
Timeout	60000
Descrizione	

La seguente tabella contiene la lista delle attività con relative frequenze ed inoltre visualizza due T-CARD QA2 una negativa

The diagram illustrates the relationship between the 'Karte' (Card) and the 'Frequenza' (Frequency) table. The 'Karte' table has columns for 'Karte Nummer', 'Karte', 'Business Excellence/Process', 'Qualität', and 'Requisiti'. The 'Frequenza' table has columns for 'Karte', 'Business Excellence/Process', 'Qualität', 'Requisiti', 'Frequenza', 'Risultato', 'Commento', and 'Bemerkung'. Red arrows indicate that the 'Karte' table is used to define the activities in the 'Frequenza' table, and that the 'Frequenza' table is used to define the activities in the 'Karte' table. The 'Frequenza' table is also used to define the activities in the 'Karte' table.

S3 – TPM / 5S assessment

Riferito ad un timeslot="S"

mese possibilità di scorrere
default mese attuale

Gemba Rundgang

Nome	1	2	5	6	7	8	9	12	13	14	15	16	19	20	21	22	23	26	27	28	29	30
Rita Özant									C3												C3	
Baran İceniz		FP54	QA3		C2	FP55	FP54	QA3				FP54	QA3		C2	FP55	FP54	QA3				
Özkan Mandıyar								QA4										QA4				
Marco Pflaier																						
Uta Möller		Safety1	FP51		Safety1	Safety1	Safety1	FP51				Safety1	FP51		Safety1	Safety1	Safety1	FP51				
Stefan	FP52					FP52										FP52						FP52
Oliver Engelmann			FP53					FP53										FP53				FP53
Alexandra Stiller		QA1	QA2					QA1	QA2									QA1	QA2			

Personale coinvolto
obtiene codice per
tipo="A"

Tutte le attività
vengono riportate
a calendario in base
alla frequenza definita
nel foglio Excel

E' possibile fare click
su una voce e creare
una nuova CARD per
questa voce

La card ha un esito
positivo o negativo
(iter approve 1/0)

Instructions:

- * For each day of the month, pick the appropriate kami shabai card.
- * Using the card, read the discipline and audit the website where the criteria should be followed. Speak to those team members you feel appropriate to determine if the criteria is met.
- * Score the audit on the schedule with either a "2" for MEETS or a "0" for DOES NOT MEET and return the card to the holder.
- * Actual is the square on the schedule below that is completed and entering your score.
- * If you scored the area "0", complete a follow up sheet and give it to the person you feel is best suited to resolve the outstanding items from the audit. Place a copy of the follow-up sheet in the slot beneath the audit schedule.
- * Re-audit the same area within 24-48 hours to see if actions have been taken.

Ein Weg, um Veränderungen zu stabilisieren

- visionare l'elenco delle card già create
- creare una nuova card con un esito

- Business excellence
- Requisiti
- Chiarimento/Spiegazione
- Prova di conformità
- Frequenza
- Responsabile
- Valutazione

Tutti questi campi andranno inseriti in project.

Per la procedura di gestione dell'informazione e la calendarizzazione sarà svolta come al seguente punto

- **Short time TPM55** nel sottoparagrafo degli **eventi periodici**

Ogni T-CARD corrisponderà ad una specifica ITER che può essere approved (meet=1) oppure rejected (meet=0), nel primo caso l'attività si chiude nel secondo si può creare una nuova card per lo stesso specifico evento.

Per poter creare una nuova T-CARD (iter) per uno specifico evento sarà necessario allocarla con un tm che ricada nel medesimo giorno, ma che sia maggiore della ITER precedente.

In base a quanto descritto aggiungiamo i seguenti campi alla struttura project.

```
ALTER TABLE project ADD COLUMN buex VARCHAR(1024), ADD COLUMN req VARCHAR(1024),
ADD COLUMN spie VARCHAR(1024), ADD COLUMN cotest VARCHAR(1024), ADD COLUMN valdesc
VARCHAR(1024), ADD COLUMN tipocar VARCHAR(101), ADD COLUMN idcar INTEGER;
comment ON COLUMN project.buex IS 'Business excellence (vedi tabella kami shabai)';
comment ON COLUMN project.req IS 'Requisiti (vedi tabella kami shabai)';
comment ON COLUMN project.spie IS 'Chiarimento/Spiegazione (vedi tabella kami
shabai)';
comment ON COLUMN project.cotest IS 'Test di conformità (vedi tabella kami
shabai)';
comment ON COLUMN project.valdesc IS 'Valutazione (vedi tabella kami shabai)';
comment ON COLUMN project.tipocar IS 'Tipo carta (vedi tabella kami shabai)';
comment ON COLUMN project.idcar IS 'Numero progressivo della carta (vedi tabella
kami shabai)';
```

Query per aggiungere le anags di tipocard per il Kami Shabai:

```
DELETE FROM anags WHERE tag='ks-tipocard';
INSERT INTO public.anags VALUES ('it', 'ks-tipocard', 'QA', 'QA', 1, '', '');
INSERT INTO public.anags VALUES ('it', 'ks-tipocard', 'C', 'C', 2, '', '');
INSERT INTO public.anags VALUES ('it', 'ks-tipocard', 'FPS', 'FPS', 3, '', '');
INSERT INTO public.anags VALUES ('it', 'ks-tipocard', 'Safety', 'Safety', 4, '',
'');
INSERT INTO public.anags VALUES ('en', 'ks-tipocard', 'QA', 'QA', 1, '', '');
INSERT INTO public.anags VALUES ('en', 'ks-tipocard', 'C', 'C', 2, '', '');
INSERT INTO public.anags VALUES ('en', 'ks-tipocard', 'FPS', 'FPS', 3, '', '');
INSERT INTO public.anags VALUES ('en', 'ks-tipocard', 'Safety', 'Safety', 4, '',
'');
INSERT INTO public.anags VALUES ('de', 'ks-tipocard', 'QA', 'QA', 1, '', '');
INSERT INTO public.anags VALUES ('de', 'ks-tipocard', 'C', 'C', 2, '', '');
INSERT INTO public.anags VALUES ('de', 'ks-tipocard', 'FPS', 'FPS', 3, '', '');
INSERT INTO public.anags VALUES ('de', 'ks-tipocard', 'Safety', 'Safety', 4, '',
'');
INSERT INTO public.anags VALUES ('pl', 'ks-tipocard', 'QA', 'QA', 1, '', '');
INSERT INTO public.anags VALUES ('pl', 'ks-tipocard', 'C', 'C', 2, '', '');
INSERT INTO public.anags VALUES ('pl', 'ks-tipocard', 'FPS', 'FPS', 3, '', '');
INSERT INTO public.anags VALUES ('pl', 'ks-tipocard', 'Safety', 'Safety', 4, '',
'');
```

S4 - Audit

Stessa struttura usata per gli AUDIT, ma solo in visualizzazione, non è possibile modificare.

Si parte da un codice plant
 si estraggono i team-board (tiposet='R')
 e si prendono gli ultimi due eventi del periodo

.. always the 2 newest audits per Team board!			
---	--	--	--

Meeting Standard

FRANKE		Meeting Standard		FPS	
ORGANISATION					
Description: Team Board		Purpose: Follow up and escalation of the performance of the day before with continuous improvement of the gaps.		Participants: Leader: Operations Mngr Timekeeper: Tbd Planning mgr, QM, team leaders, Safety Manager, Maintenance mngr, Operations manager, Engineering.	
Location: FBS Line 1					
Frequency: Daily					
Day: Mo/Tu/We/Th/Fr					
Time: 12:15					
INFORMATIONS					
Key Performance Indicator (KPI's): Upload		Warnings Upload		Output of Meeting: Upload	
Safety				Action plan	
Quality					
Delivery		COVID new regulation			
Cost					
PROCESS					
Agenda:		Responsible:		Time:	
1. Opening		Leader		0,5 min	
2. VSM 1 - SQCD->Actions				2 min	
3. VSM 2 - SQDC->Actions				2 min	
4. VSM 3 - SQDC->Actions				2 min	
5. Quality - External complaints		Quality Manager		3 min	
6. Maintenance		Maintenance Manager		2 min	
7. Engineering				3 min	
8. Closing		Leader		0,5 min	

```

CREATE TABLE public.protocol (
    codice INTEGER NOT NULL,
    codubi CHARACTER VARYING(51),
    frequency CHARACTER VARYING(51),
    DAY CHARACTER VARYING(101),
    TIME CHARACTER VARYING(10),
    purpose CHARACTER VARYING(3000),
    actionplan CHARACTER VARYING(3000),
    leader CHARACTER VARYING(101),
    timekeeper CHARACTER VARYING(101),
    partecipanti CHARACTER VARYING(3000),
    warnimage CHARACTER VARYING(101)
);

ALTER TABLE public.protocol OWNER TO postgres;
COMMENT ON TABLE public.protocol IS 'Tabella dei Meeting Protocols';
COMMENT ON COLUMN public.protocol.codubi IS 'Codice del luogo';
COMMENT ON COLUMN public.protocol.frequency IS 'Frequenza del meeting';
COMMENT ON COLUMN public.protocol.day IS 'Giorni della settimana del meeting';
COMMENT ON COLUMN public.protocol.time IS 'Orario del meeting';
COMMENT ON COLUMN public.protocol.purpose IS 'Scopo del meeting';
COMMENT ON COLUMN public.protocol.actionplan IS 'Actionplan del meeting';
COMMENT ON COLUMN public.protocol.leader IS 'Leader del meeting';
COMMENT ON COLUMN public.protocol.timekeeper IS 'Timekeeper del meeting';
COMMENT ON COLUMN public.protocol.partecipanti IS 'Partecipanti al meeting';
COMMENT ON COLUMN public.protocol.warnimage IS 'Immagine warning di anteprima';
ALTER TABLE ONLY public.protocol ADD CONSTRAINT ix_protocol_codice PRIMARY KEY (codice);
ALTER TABLE ONLY public.protocol ADD CONSTRAINT ix_protocol_codubi FOREIGN KEY (codubi) REFERENCES public.ubicaz(codice) ON UPDATE CASCADE ON DELETE RESTRICT;

```

```
CREATE TABLE public.agenda (  
    codprot INTEGER NOT NULL,  
    pos INTEGER,  
    descrizio CHARACTER VARYING(256),  
    responsabile CHARACTER VARYING(101),  
    TIME FLOAT  
);  
ALTER TABLE public.agenda OWNER TO postgres;  
COMMENT ON TABLE public.agenda IS 'Tabella agenda per i meeting protocols';  
COMMENT ON COLUMN public.agenda.codprot IS 'Codice riferimento protocollo';  
COMMENT ON COLUMN public.agenda.pos IS 'Ordine degli argomenti in agenda';  
COMMENT ON COLUMN public.agenda.descrizio IS 'Descrizione';  
COMMENT ON COLUMN public.agenda.responsabile IS 'Responsabile - ruolo che ricopre';  
COMMENT ON COLUMN public.agenda.time IS 'durata del punto in agenda';  
ALTER TABLE ONLY public.agenda ADD CONSTRAINT ix_agenda_codprot FOREIGN KEY  
(codprot) REFERENCES public.protocol(codice) ON UPDATE CASCADE ON DELETE CASCADE;  
  
CREATE TABLE public.protrules (  
    codprot INTEGER NOT NULL,  
    pos INTEGER,  
    descrizio CHARACTER VARYING(256)  
);  
ALTER TABLE public.protrules OWNER TO postgres;  
COMMENT ON TABLE public.protrules IS 'Tabella agenda per i meeting protocols';  
COMMENT ON COLUMN public.protrules.codprot IS 'Codice riferimento protocollo';  
COMMENT ON COLUMN public.protrules.pos IS 'Ordine degli argomenti in agenda';  
COMMENT ON COLUMN public.protrules.descrizio IS 'Descrizione';  
ALTER TABLE ONLY public.protrules ADD CONSTRAINT ix_protrules_codprot FOREIGN KEY  
(codprot) REFERENCES public.protocol(codice) ON UPDATE CASCADE ON DELETE CASCADE;
```

Aggiunta dei campi al meeting protocol

```
ALTER TABLE protocol ADD COLUMN kpisafety VARCHAR(51);  
ALTER TABLE protocol ADD COLUMN kpiquality VARCHAR(51);  
ALTER TABLE protocol ADD COLUMN kpidelivery VARCHAR(51);  
ALTER TABLE protocol ADD COLUMN kpicost VARCHAR(51);  
ALTER TABLE protocol ADD COLUMN warnimage VARCHAR(101);
```

Safety Report

Report LOST TIME ACCIDENT or NEAR MISS				
#	Finalized	Item	Description	Who
1	dd.mm.yyyy	Accident Description	An employee received an injury to the thumb of his right hand while working at the workplace.	Mr. Zoff
2	dd.mm.yyyy	Interim Containment Actions	Inform the employees about the accident and that they must carry out their work with due caution. Especially using the foreseen safety equipment.	Mr S. Mazzola
3	dd.mm.yyyy	Root Cause Analysis	Interview with employee done.	Mr. G. Meazza
4	dd.mm.yyyy	Preventive Actions	Check of the safety equipment - done.	K.Klimiuk
5	dd.mm.yyyy	Verifications of Corrective Actions	open	Mr. Dziadura (work safety specialist)
6	dd.mm.yyyy	Lost Time (days)	You can only close the report , if lost time is reported	
7	dd.mm.yyyy	Report Closed	Lead Time = Date 1-Date 6	Mr. P. Rossi

Picture

```

CREATE TABLE public.safetyreport (
    codice INTEGER NOT NULL,
    descrizio CHARACTER VARYING(101),
    zord INTEGER,
    tipodato VARCHAR(2)
);

ALTER TABLE public.safetyreport OWNER TO postgres;
COMMENT ON TABLE public.safetyreport IS 'Tabella tematica dei safety report';
ALTER TABLE ONLY public.safetyreport ADD CONSTRAINT ix_safetyreport_codice PRIMARY
KEY (codice);

INSERT INTO safetyreport (codice, descrizio, zord) VALUES (1, 'Accident
Description', 1);
INSERT INTO safetyreport (codice, descrizio, zord) VALUES (2, 'Interim Containment
Actions', 2);
INSERT INTO safetyreport (codice, descrizio, zord) VALUES (3, 'Root Cause
Analysis', 3);
INSERT INTO safetyreport (codice, descrizio, zord) VALUES (4, 'Preventive Actions',
4);
INSERT INTO safetyreport (codice, descrizio, zord) VALUES (5, 'Verifications of
Corrective Actions', 5);
INSERT INTO safetyreport (codice, descrizio, zord) VALUES (6, 'Lost Time (days)',
6);
INSERT INTO safetyreport (codice, descrizio, zord) VALUES (7, 'Report Closed', 7);

CREATE TABLE public.safprj (
    codsafety INTEGER,
    codprj INTEGER,
    codoper VARCHAR(51),
    descrizio VARCHAR(512),

```

```

tmfin FLOAT
);
ALTER TABLE public.safprj OWNER TO postgres;
COMMENT ON TABLE public.safprj IS 'Associazione i project di tipo safety e i temi
del safety report';

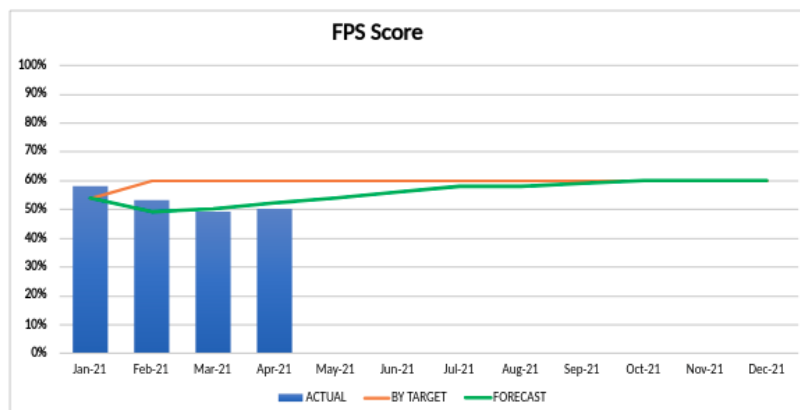
ALTER TABLE ONLY public.safprj ADD CONSTRAINT ix_safprj_codsafety FOREIGN KEY
(codsafety) REFERENCES public.safetyreport(codice) ON UPDATE CASCADE ON DELETE
RESTRICT;
ALTER TABLE ONLY public.safprj ADD CONSTRAINT ix_safprj_codprj FOREIGN KEY (codprj)
REFERENCES
public.project(codice) ON UPDATE CASCADE ON DELETE CASCADE;
ALTER TABLE ONLY public.safprj ADD CONSTRAINT ix_safprj_codoper FOREIGN KEY
(codoper) REFERENCES
public.operato(codice) ON UPDATE CASCADE ON DELETE RESTRICT;

```

FPS ACHIVEMENT

..

Month	ACTUAL	BY TARGET	FORECAST
January-21	58%	54%	54%
February-21	53%	60%	49%
March-21	49%	60%	50%
April-21	50%	60%	52%
May-21		60%	54%
June-21		60%	56%
July-21		60%	58%
August-21		60%	58%
September-21		60%	59%
October-21		60%	60%
November-21		60%	60%
December-21		60%	60%



```

CREATE TABLE fpsachievement (
    codubi CHARACTER VARYING(51),
    MONTH INTEGER,
    YEAR INTEGER,
    actual FLOAT,
    target FLOAT,
    forecast FLOAT
);

ALTER TABLE ONLY public.fpsachievement ADD CONSTRAINT ix_fpsachievement_pkey
PRIMARY KEY (codubi, MONTH, YEAR);
ALTER TABLE ONLY public.fpsachievement ADD CONSTRAINT ix_fpsachievement_codubi
FOREIGN KEY (codubi) REFERENCES public.ubicaz(codice) ON UPDATE CASCADE ON DELETE
CASCADE;

```

Inserimento manuale dati per le risorse che non sono connesse con sistema OEE Next

Per alcune teamboards è stata data la possibilità di aggiungere manualmente dei KPIs relativi su qualità, costo e

delivery. In particolare per "Scraps & Reworks Pareto", "Downtime Pareto" e "Order Achivement Pareto". Questa tabella per gestire gli inserimenti manuali:

```
CREATE TABLE public.kpispareto (
  codubi CHARACTER VARYING(101),
  causa CHARACTER VARYING(101),
  tm DOUBLE PRECISION,
  nr INTEGER,
  codoper CHARACTER VARYING(101),
  TYPE CHARACTER VARYING(5),
  DATA CHARACTER VARYING(9),
  codice INTEGER NOT NULL
);
COMMENT ON COLUMN public.kpispareto.tm IS 'tm di registrazione';
COMMENT ON COLUMN public.kpispareto.type IS 'D = Downtime Pareto, S = Scraps & Rework pareto, O = Order Achivement Pareto';
COMMENT ON COLUMN public.kpispareto.data IS 'Data di registrazione';
ALTER TABLE ONLY public.kpispareto ADD CONSTRAINT ix_kpispareto_pk PRIMARY KEY (codice);
ALTER TABLE ONLY public.kpispareto ADD CONSTRAINT ix_kpispareto_codoper FOREIGN KEY (codoper) REFERENCES public.operato(codice) ON UPDATE CASCADE ON DELETE CASCADE;
ALTER TABLE ONLY public.kpispareto ADD CONSTRAINT ix_kpispareto_codubi FOREIGN KEY (codubi) REFERENCES public.ubicaz(codice) ON UPDATE CASCADE ON DELETE CASCADE;
```

Per abilitare quali risorse dovrenno utilizzare la pagina di inserimento manuale è stato messo su wip-settings la seguente chiave:

```
manualkpis:{
  list:[
    {k:'150121', v:'Others Fabricated (DAN)'},
    {k:'FIMDIP03', v:'TB Fabricated Upstream'},
    {k:'FIMDIP04', v:'TB Fabricated Downstream'},
  ]
}
```

I codici presenti in lista avranno accesso alla pagina di inserimento manuale dei dati nei pareto. Le altre risorse invece accederanno alla vista di default, con i dati collegati a OEE

Proactive KPIs

Standard KPI in Cognos Extraction	
1. # of lost time accidents (severe/not severe): Cognos	_____
2. Accident Frequency Rate: Cognos	_____
3. Accident Severity Rate: Cognos	_____
4. 5S score	_____
5. # of deviations/ Gemba Safety visits	_____
6. Maturity Assessment score (1-5)	_____
7. # of actions closed < 3 months and actions open > 3 months	_____

AgentGraphSup configurazione

```
<agent lib="/usr/lib/libdigcic.so.0.0.0" create="agentGraphSup" live="1"
```

```
name="graph">
  <param name="agps" value="prepstat"/>
</agent>
```

Safety Actions

Per gestire alcuni KPIs sulle actions abbiamo dovuto introdurre il concetto di "Safety Action".

Una Safety Action è un action generica che può essere associata a diversi tipi di oggetto (objiter), in particolare a:

- Job Safety Observations → objiter.tipo='H'
- Safety Report → objiter.tipo='A|N|B|F|R'
- Risk Assessmebr → objiter.tipo='U' (nuovo oggetto che può essere gestito in modo simile a un JSO)

Per implementare utilizzando i dati che storicamente sono stati inseriti, vengono trasformate in "safety actions" i seguenti oggetti per tipo:

Job Safety Observation → ogni suggestion di una JSO diventa una safety action. Le suggestion dei JSO sono salvate nella tabella prjunsafe

Safety Report → nel safety report ci sono configurare 2 action (safprj.codsafety = 2 e safprj.codsafey = 4).

Ognuna di queste action diventano una safety action.

Struttura di una safety action:

```
CREATE TABLE public.actionstype (
    codice INTEGER NOT NULL,
    descrizio CHARACTER VARYING(51),
    tipo CHARACTER VARYING(51),
    zord INTEGER
);
ALTER TABLE public.actionstype OWNER TO postgres;
COMMENT ON TABLE public.actionstype IS 'Tabella che contiene la tipologia delle azioni legate al safety (Safety Report, Job Safety Observation, Risk)';
COMMENT ON COLUMN public.actionstype.tipo IS 'Tipologia di tipi di azione. RA -> Risk Assessment, SR -> Safety Report & JSO';
ALTER TABLE ONLY public.actionstype ADD CONSTRAINT ix_actionstype_pk PRIMARY KEY (codice);

INSERT INTO actionstype (codice, descrizio, tipo, zord) VALUES (1, 'Azioni di contenimento provvisorie', 'SR', 1);
INSERT INTO actionstype (codice, descrizio, tipo, zord) VALUES (2, 'Azioni preventive', 'SR', 2);

INSERT INTO actionstype (codice, descrizio, tipo, zord) VALUES (3, 'Building', 'RA', 1);
INSERT INTO actionstype (codice, descrizio, tipo, zord) VALUES (4, 'Installation', 'RA', 2);
INSERT INTO actionstype (codice, descrizio, tipo, zord) VALUES (5, 'Zoning', 'RA', 3);
INSERT INTO actionstype (codice, descrizio, tipo, zord) VALUES (6, 'PPE', 'RA', 4);
INSERT INTO actionstype (codice, descrizio, tipo, zord) VALUES (7, 'Tools', 'RA', 5);
INSERT INTO actionstype (codice, descrizio, tipo, zord) VALUES (8, 'Another', 'RA', 6);
```

```

CREATE TABLE public.safetyactions (
    codprj INTEGER,
    descrizio CHARACTER VARYING(1000),
    tmcre DOUBLE PRECISION,
    tmtarget DOUBLE PRECISION,
    tmclosed DOUBLE PRECISION,
    acttype INTEGER,
    STATUS CHARACTER VARYING(2),
    codoper CHARACTER VARYING(51),
    zord INTEGER
);

ALTER TABLE public.safetyactions OWNER TO postgres;
COMMENT ON TABLE public.safetyactions IS 'Tabella che contiene tutte le azioni preventive e correttive legate al safety (Safety Report, Job Safety Observation, Risk)';
COMMENT ON COLUMN public.safetyactions.codprj IS 'project.codice';
COMMENT ON COLUMN public.safetyactions.tmcre IS 'tm di creazione dell''azione';
COMMENT ON COLUMN public.safetyactions.tmtarget IS 'tm di prevista chiusura azione';
COMMENT ON COLUMN public.safetyactions.tmclosed IS 'tm chiusura effettiva dell''azione';
COMMENT ON COLUMN public.safetyactions.status IS '0 = Open, C = Close';
COMMENT ON COLUMN public.safetyactions.zord IS 'Posizione';
ALTER TABLE ONLY public.safetyactions ADD CONSTRAINT ix_safetyactions_acttype FOREIGN KEY (acttype) REFERENCES public.actionstype(codice) ON UPDATE CASCADE ON DELETE RESTRICT;
ALTER TABLE ONLY public.safetyactions ADD CONSTRAINT ix_safetyactions_codoper FOREIGN KEY (codoper) REFERENCES public.operato(codice) ON UPDATE CASCADE ON DELETE CASCADE;
ALTER TABLE ONLY public.safetyactions ADD CONSTRAINT ix_safetyactions_codprj FOREIGN KEY (codprj) REFERENCES public.project(codice) ON UPDATE CASCADE ON DELETE CASCADE;

```

Query per modificare e trasformare in safety actions le informazioni dei JSO e dei Safety Report esistenti:

```

INSERT INTO safetyactions SELECT safprj.codprj, safprj.descrizio, objiter.tmcre AS tmcre, tmfin AS tmtarget, tmfeed AS tmclosed, '1' AS acttype, 'C' AS STATUS, safprj.codoper, '1' AS zord FROM safprj JOIN objiter ON safprj.codprj = objiter.codice JOIN iter ON objiter.codice = iter.codobj WHERE objiter.state = 'A' AND safprj.codsafety = 2;

```

```

INSERT INTO safetyactions SELECT safprj.codprj, safprj.descrizio, objiter.tmcre AS tmcre, tmfin AS tmtarget, NULL AS tmclosed, '1' AS acttype, '0' AS STATUS, safprj.codoper, '1' AS zord FROM safprj JOIN objiter ON safprj.codprj = objiter.codice JOIN iter ON objiter.codice = iter.codobj WHERE objiter.state <> 'A' AND safprj.codsafety = 2;

```

```

INSERT INTO safetyactions SELECT safprj.codprj, safprj.descrizio, objiter.tmcre AS tmcre, tmfin AS tmtarget, tmfeed AS tmclosed, '2' AS acttype, 'C' AS STATUS, safprj.codoper, '1' AS zord FROM safprj JOIN objiter ON safprj.codprj = objiter.codice JOIN iter ON objiter.codice = iter.codobj WHERE objiter.state = 'A' AND safprj.codsafety = 4;

```

```

INSERT INTO safetyactions SELECT safprj.codprj, safprj.descrizio, objiter.tmcre AS tmcre, tmfin AS tmtarget, NULL AS tmclosed, '2' AS acttype, '0' AS STATUS,

```

```
safprj.codoper, '1' AS zord FROM safprj JOIN objiter ON safprj.codprj =  
objiter.codice JOIN iter ON objiter.codice = iter.codobj WHERE objiter.state <> 'A'  
AND safprj.codsafety = 4;
```

```
DELETE FROM safprj WHERE codsafety = 2;  
DELETE FROM safetyreport WHERE codice = 2;  
DELETE FROM safprj WHERE codsafety = 4;  
DELETE FROM safetyreport WHERE codice = 4;  
UPDATE safetyreport SET codice = 2, zord = 2 WHERE codice = 3;  
UPDATE safetyreport SET codice = 3, zord = 3 WHERE codice = 5;  
UPDATE safetyreport SET codice = 4, zord = 4 WHERE codice = 6;  
UPDATE safetyreport SET codice = 5, zord = 5 WHERE codice = 7;
```

```
INSERT INTO safetyactions SELECT prjunsafe.codprj, prjunsafe.descrizio,  
objiter.tmcre, NULL AS tmtarget, tmfeed AS tmclosed, NULL AS acttype, 'C' AS  
STATUS, NULL AS codoper, pos AS zord FROM prjunsafe JOIN objiter ON  
prjunsafe.codprj = objiter.codice LEFT JOIN iter ON objiter.codice = iter.codobj  
WHERE objiter.tipo = 'H' AND objiter.state = 'A';  
DELETE FROM objiter WHERE codice = 15456;  
INSERT INTO safetyactions SELECT prjunsafe.codprj, prjunsafe.descrizio,  
objiter.tmcre, NULL AS tmtarget, NULL AS tmclosed, NULL AS acttype, 'O' AS STATUS,  
NULL AS codoper, pos AS zord FROM prjunsafe JOIN objiter ON prjunsafe.codprj =  
objiter.codice LEFT JOIN iter ON objiter.codice = iter.codobj WHERE objiter.tipo =  
'H' AND objiter.state <> 'A';
```

KPI #4 - 5S Score

5S score

Purpose:

The 5S Rate % emphasizes Franke's commitment to apply the 5S methodology to ensure order, productivity, and decrease safety risk in all working areas of the company.

Context:

In a workplace where there are significant hazards, and safety is a real concern, it makes a lot of sense to follow 5S so safety can get the attention that is needed. This essential step of 5S focuses on identifying hazards and setting preventive controls to keep workers safe during work operations and ensure that the work environment meets required safety standards.

Metrics:

$$\text{5S Score} = \frac{\sum (\text{5S Audit checklist score per area} \times \text{Total number of employees work at that area})}{\text{Total number of employees work at all assessed areas}}$$

Example: How to calculate the 5S score of a single area shown below at the checklist;

$$\text{5S Score} = (3+3+2+1+1)/5 = 2$$

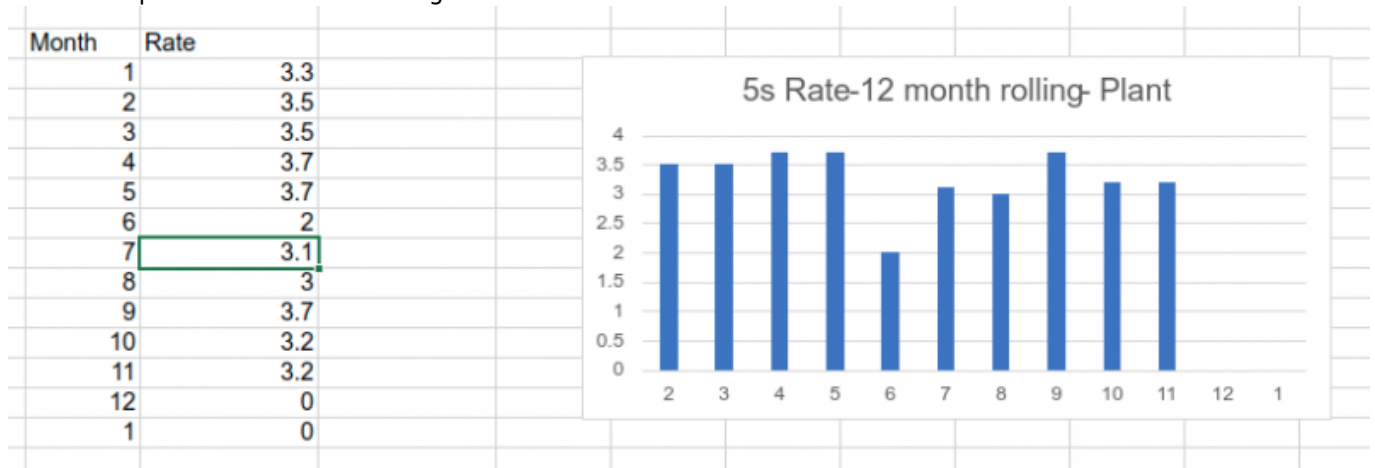
Example: How to calculate the 5S Score of 2 areas;

If there are 8 employees working in one area with a 5S score of 2 and 2 employees working at another area with a 5S score of 1, then over all 5S score is calculated by;

$$\text{5S Score} = \frac{(8 \times 2 + 2 \times 1)}{(8 + 2)} = 1,8$$

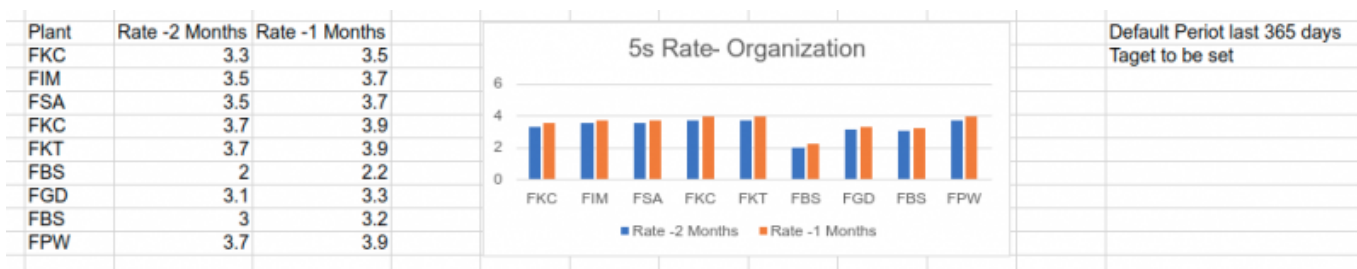
1 - Plant Level:

Come periodo di riferimento si prende gli ultimi 365 giorni (in 12 mesi). Il plant rate è una media pesata per il numero di persone che lavora ad ogni asset dei vari 5s audit fatti.



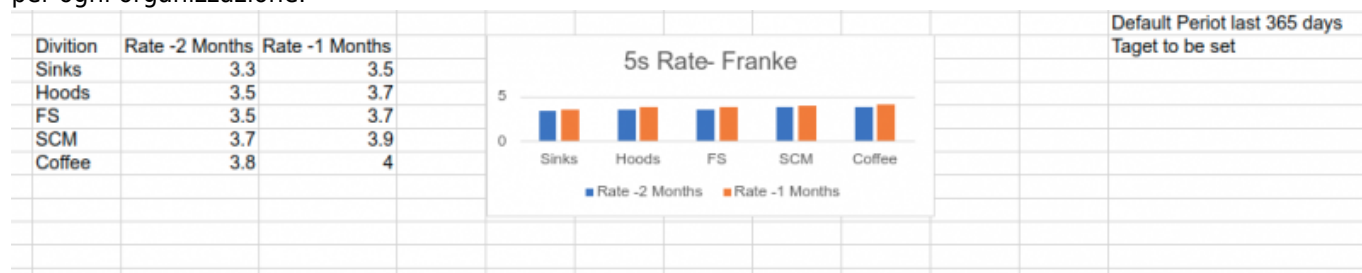
2 - Organization Level:

Come periodo di riferimento si prende gli ultimi 365 giorni. Viene mostrato il plant rate medio del periodo selezionato, data una singola organizzazione.



3 - Franke Level:

Come periodo di riferimento si prende gli ultimi 365 giorni. Viene mostrata la media di tutti i plant rate medio per ogni organizzazione.



⚠ NB I target devono essere messi a livello di ubicaz e rappresenterà una linea orizzontale sopra il grafico.

Query per estrarre lo score (media voto delle 5 risposte dell'audit) del singolo audit. Ogni riga corrisponde ad un audit con il suo score medio. Se il codubi dell'audit è di tiposet 'P' allora abbiamo in mancost il numero di operatori che lavorano su quell'asset. Questo mancost ci servirà per fare la media pesata.

```
WITH RECURSIVE treeubi AS (SELECT u.* FROM ubicaz u WHERE u.codice = 'FKT' UNION
ALL SELECT u.* FROM ubicaz u, treeubi WHERE u.codubib = treeubi.codice),
sinans AS (SELECT codoper, MAX(codice) AS codice, codclista FROM answer GROUP BY
codoper, codclista)
SELECT objiter.codice, objiter.tmcre, objiter.tipo, objiter.dayofmonth,
objiter.month, objiter.year, project.descrizio, operato.nominati AS creator,
treeubi.descrizio AS asset, treeubi.tiposet, treeubi.mancost, project.codubi,
AVG(answer.value::NUMERIC) AS score
FROM objiter INNER JOIN project ON objiter.codice = project.codice INNER JOIN
operato ON objiter.codoper = operato.codice INNER JOIN treeubi ON project.codubi =
treeubi.codice
JOIN prjtask ON prjtask.codprj = objiter.codice JOIN clista ON clista.codptask =
prjtask.codice JOIN sinans ON clista.codice = sinans.codclista JOIN answer ON
sinans.codice = answer.codice
WHERE objiter.tipo='P' AND objiter.tmcre >= 1672527600 AND objiter.tmcre <=
1704063599
GROUP BY objiter.codice, objiter.tmcre, objiter.tipo, objiter.dayofmonth,
objiter.month, objiter.year, project.descrizio, creator, asset, tiposet, mancost,
project.codubi
ORDER BY tmcre DESC
```

Dato che in precedenza gli audit venivano fatti a livello di teamboard (tiposet 'R') e questa specifica è cambiata negli ultimi due mesi proprio per gestire il numero di operatori per asset, è possibile che se un audit è stato precedentemente fatto a livello di teamboard non abbia mancost per fare la media pesata. In questo caso potremmo gestire ciò (a tendere non ci saranno più audit a livello di teamboard ma solo a livello di asset) sommando tutti gli operatori di tutti gli asset di ogni teamboard e facendo la media pesata con questo valore.

```
SELECT t.codice, SUM(a.mancost) FROM ubicaz t JOIN ubicaz a ON t.codice = a.codubib
WHERE t.tiposet = 'R' GROUP BY t.codice
```

! QUERY PER I GRAFICI

Questa query prende in ingresso un codubi (codice del plant), un fromtm ed un totm. I tm vengono utilizzati come filtro sul cambo objiter.tmcre, che dovrà essere incluso tra i due tm passati. Per default i due tm devono essere gli ultimi 12 mesi, escluso il mese corrente. Esempio, sono a gennaio 2024, il fromtm deve essere il 01/01/2023 ed il totm 31/12/2023

```
WITH RECURSIVE treeubi AS (SELECT u.* FROM ubicaz u WHERE u.codice = 'FKT' UNION
ALL SELECT u.* FROM ubicaz u, treeubi WHERE u.codubib = treeubi.codice),
sinans AS (SELECT codoper, MAX(codice) AS codice, codclista FROM answer GROUP BY
codoper, codclista),
scores AS (SELECT objiter.codice, objiter.tmcre, objiter.tipo, objiter.dayofmonth,
objiter.month, objiter.year, project.descrizio, operato.nominati AS creator,
treeubi.descrizio AS asset, treeubi.tiposet, treeubi.mancost, project.codubi,
AVG(answer.value::NUMERIC) AS score
FROM objiter INNER JOIN project ON objiter.codice = project.codice INNER JOIN
operato ON objiter.codoper = operato.codice INNER JOIN treeubi ON project.codubi =
treeubi.codice
JOIN prjtask ON prjtask.codprj = objiter.codice JOIN clista ON clista.codptask =
prjtask.codice JOIN sinans ON clista.codice = sinans.codclista JOIN answer ON
sinans.codice = answer.codice
WHERE objiter.tipo='P' AND objiter.tmcre >= 1676502000 AND objiter.tmcre <=
1705391447 AND mancost IS NOT NULL AND mancost > 0
GROUP BY objiter.codice, objiter.tmcre, objiter.tipo, objiter.dayofmonth,
objiter.month, objiter.year, project.descrizio, creator, asset, tiposet, mancost,
project.codubi
ORDER BY tmcre DESC),
totoper AS (SELECT SUM(mancost) AS total, MONTH, YEAR FROM scores GROUP BY MONTH,
YEAR)
SELECT 'FKT' AS resource, scores.year, scores.month AS xasys,
SUM((mancost*score))/totoper.total AS v_fivesrate FROM scores JOIN totoper ON
scores.year = totoper.year AND scores.month = totoper.month
GROUP BY scores.year, scores.month, totoper.total, resource
ORDER BY resource, scores.year DESC, scores.month DESC
```

Questa query, per FKT, ritorna per ogni mese il 5s rating, che è proprio il valore che serve a disegnare il grafico:

resource	year	month	fivesrate
FKT	2023	11	2.2175000000000002
FKT	2023	10	2.1333333333333333
FKT	2023	9	1.8

La seguente query è sufficiente sia per l'organization che per franke level, basta passare un codubi di organization o il codubi di franke

```
WITH RECURSIVE treeubi AS (
SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.tiposet, ubicaz.codubib,
ARRAY[]::VARCHAR[] || ubicaz.descrizio AS ancestors, ARRAY[]::VARCHAR[] ||
ubicaz.codice AS ancestorscode, ARRAY[]::VARCHAR[] || ubicaz.tiposet AS
ancestorsts, mancost
FROM ubicaz WHERE codice = 'franke001'
UNION ALL
SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.tiposet, ubicaz.codubib,
treeubi.ancestors || ubicaz.descrizio, treeubi.ancestorscode || ubicaz.codice,
treeubi.ancestorsts || ubicaz.tiposet, ubicaz.mancost
FROM ubicaz, treeubi
WHERE ubicaz.codubib = treeubi.codice
```



```

),
sinans AS (
  SELECT codoper, MAX(codice) AS codice, codclista
  FROM answer
  GROUP BY codoper, codclista
),
scores AS (
  SELECT ancestors, ancestorscode, objiter.codice, objiter.tmcre, objiter.tipo,
  objiter.dayofmonth, objiter.month, objiter.year, project.descrizio,
  operato.nominati AS creator, treeubi.descrizio AS asset, treeubi.tiposet,
  treeubi.mancost, project.codubi, AVG(answer.value::NUMERIC) AS score
  FROM objiter
  INNER JOIN project ON objiter.codice = project.codice
  INNER JOIN operato ON objiter.codoper = operato.codice
  INNER JOIN treeubi ON project.codubi = treeubi.codice
  JOIN prjtask ON prjtask.codprj = objiter.codice
  JOIN clista ON clista.codptask = prjtask.codice
  JOIN sinans ON clista.codice = sinans.codclista
  JOIN answer ON sinans.codice = answer.codice
  WHERE objiter.tipo='P'
  AND mancost IS NOT NULL
  AND mancost > 0
  AND objiter.tmcre >= 1676502000
  AND objiter.tmcre <= 1705391447
  GROUP BY ancestors, ancestorscode, objiter.codice, objiter.tmcre, objiter.tipo,
  objiter.dayofmonth, objiter.month, objiter.year, project.descrizio, creator, asset,
  tiposet, mancost, project.codubi
  ORDER BY tmcre DESC
),
totoper AS (
  SELECT SUM(mancost) AS total, MONTH, YEAR
  FROM scores
  GROUP BY MONTH, YEAR
)
SELECT REPLACE(REPLACE(SPLIT_PART(ancestors::text, ',', 2), '{', ''), ' ', '') AS
resource, REPLACE(SPLIT_PART(ancestorscode::text, ',', 2), '{', '') AS xasys,
scores.year, scores.month, SUM((mancost*score))/totoper.total AS v_fivesrate
FROM scores
JOIN totoper ON scores.year = totoper.year AND scores.month = totoper.month
GROUP BY scores.year, scores.month, totoper.total, resource, xasys
ORDER BY resource, scores.year DESC, scores.month DESC;

```

Chiamata Agent KPI #4

PLANT level:

- Nome dell'agente: graph
 - ask
 - request: fivescore
 - p1: rec(codubi,fromtm,totm,tiposet)
 - codubi = codice ubicaz
 - fromtm = tm di inizio
 - totm = tm di fine
 - tiposet = tipo settore (S)
 - retv: rec(resource,year,xasys,v_fivesrate) ESEMPIO:
 - resource = "FKT"
 - year = 2023

- xasys = 12
- v_fivesrate = 2.6

ORGANIZATIONAL and FRANKE level:

- Nome dell'agente: graph
 - ask
 - request: fivescore
 - p1: rec(codubi,fromtm,totm,tiposet)
 - codubi = codice ubicaz
 - fromtm = tm di inizio
 - totm = tm di fine
 - tiposet = tipo settore (A o W)
 - retv: rec(resource,xasys,year,month,v_fivesrate) ESEMPIO:
 - resource = "CICBoardPlantFKTSwitzerlandAarburg"
 - xasys = "FKT"
 - year = 2023
 - month = 12
 - v_fivesrate = 0.2

KPI #5 - Nr. of deviations/Safety visits

Safety Observation Walk

Visit and walk around of a work location, undertaken by Managers and Supervisors to observe working practices, engage with people and talk about Health and Safety and everyday working behaviors.

- Identification of hazards and related risks, (un)safe act, and conditions
- Identification of hazards and risk by an external view
- Demonstrate management buy-in
- Coaching in both ways, visitors and local operations
- Compliment safe behavior and create commitment to change and adapt unsafe behavior/practices

Definition of walk:

Prepared Gemba walk with 1 max 3 persons observing for 5-10 minutes activity of employee. After the observation 5' talk with the employee regarding observed safe behavior and conditions = compliment and with open questions (what could happen if ? What is the standard regarding topic x etc, review observed unsafe behavior and/or conditions: result is agreement regarding defined actions/timings and commitment to improve/change behavior.

Metrics:

of walks is defined and planned based on the # of FTE and should be a minimum of 2 tours/FTE/year e.g plant with 200 FTE should plan/execute 400 tours a year = around 33/months

Average number of deviations (unsafe acts/conditions) observed per walk

Formula:

= total number of reported deviations/ # of executed walks for that month

Andrebbe aggiunto al DSFM un nuovo elemento, la gemba walk observaion. È un report che viene fatto dai manager e dai supervisori che vanno in visita nel processo. Dato che di fatto sarebbero identiche alle Job Safety

Observations, con Bladimir abbiamo concordato di aggiungere un flag agli oggetti JSO per identificare quelli che sono JSO e quelli che sono Gemba Walk Observation Report.

Per questo KPI si considereranno quindi tutti i JSO che hanno il flag "Gemba Walk Observation Report", ed in particolare le safety actions relative.

I "nr. of deviations" di un gemba walk sarebbero le azioni legate al gemba walk stesso.

```
ALTER TABLE project ADD COLUMN sogw INTEGER DEFAULT 0;
COMMENT ON COLUMN project.sogw IS '0 -> classico JSO, 1 -> Safety Observation Gemba Walk.';
```

Nuovo campo sogw nella tabella project, se 1 → Gemba Walk Observation, se 0, classico JSO. Quindi per questo KPI vanno prese in considerazione tutte e sole le action degli objiter di tipo 'H', legate ai project con sogw = 1.

SOGW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

SW 1

Le safety visits sono dati che a tendere andremo ad imporarci da un file excel che ci metteranno a disposizione ogni mese, e deriva da un'estrazione dal loro cognos ibm. Al momento diamo loro la possibilità di inserire manualmente questi valori. Comunque andranno salvati in una tabella. Le safety visits sono per mese per plant.

```
CREATE TABLE public.kpiubiext (
    codubi CHARACTER VARYING(51) NOT NULL,
    MONTH INTEGER NOT NULL,
    YEAR INTEGER NOT NULL,
    fte INTEGER
);
```

```
COMMENT ON COLUMN kpiubiext.fte IS 'il numero di executed walks per mese da parte del business';
```

```
ALTER TABLE public.kpiubiext OWNER TO postgres;
ALTER TABLE ONLY public.kpiubiext ADD CONSTRAINT ix_kpiubiext_pkey PRIMARY KEY (codubi, MONTH, YEAR);
ALTER TABLE ONLY public.kpiubiext ADD CONSTRAINT ix_kpiubiext_codubi FOREIGN KEY (codubi) REFERENCES public.ubicaz(codice) ON UPDATE CASCADE ON DELETE CASCADE;
```

! QUERY PER I GRAFICI

```
WITH RECURSIVE treeubi AS (SELECT u.* FROM ubicaz u WHERE u.codice = 'FKT' UNION ALL SELECT u.* FROM ubicaz u, treeubi WHERE u.codubib = treeubi.codice),
kpiext AS (SELECT * FROM kpiubiext WHERE codubi = 'FKT')
```

```

SELECT 'FKT' AS resource, objiter.month, objiter.year, COUNT(safetyactions.*) AS
tasks, kpiext.fte, COUNT(safetyactions.):FLOAT/fte AS rate FROM objiter JOIN
project ON objiter.codice = project.codice JOIN treeubi ON project.codubi =
treeubi.codice
JOIN safetyactions ON project.codice = safetyactions.codprj JOIN kpiext ON
objiter.month = kpiext.month AND objiter.year = kpiext.year
WHERE tipo = 'H' AND project.sogw = 1 AND objiter.tmcre >= 1676502000 AND
objiter.tmcre <= 1705391447
GROUP BY resource, objiter.month, objiter.year, fte ORDER BY YEAR DESC, MONTH DESC

```

Questa query, per FKT, ritorna per ogni mese rate del numero di actions sul numero di safety visits (kpiubiext.fte):

resource	month	year	tasks	fte	rate
FKT	1	2024	5	14	0.35714285714285715
FKT	10	2023	2	16	0.125
FKT	9	2023	9	20	0.45
FKT	8	2023	10	5	2
FKT	7	2023	1	10	0.1

- A livello di Organization sarà la somma di tutte le actions del periodo stabilito sulla somma del numero FTE del periodo stabilito (per ogni plant dell'organization)

```

WITH RECURSIVE treeubi AS (
    SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.tiposet, ubicaz.codubib,
    ARRAY[]::VARCHAR[] || ubicaz.descrizio AS ancestors, ARRAY[]::VARCHAR[] ||
    ubicaz.codice AS ancestorscode, ARRAY[]::VARCHAR[] || ubicaz.tiposet AS
    ancestorsts, mancost
    FROM ubicaz WHERE codice = 'franke001'
    UNION ALL
    SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.tiposet, ubicaz.codubib,
    treeubi.ancestors || ubicaz.descrizio, treeubi.ancestorscode || ubicaz.codice,
    treeubi.ancestorsts || ubicaz.tiposet, ubicaz.mancost
    FROM ubicaz, treeubi
    WHERE ubicaz.codubib = treeubi.codice
),

kpiext AS (SELECT kpiubiext.* FROM kpiubiext JOIN treeubi ON kpiubiext.codubi =
treeubi.codice),
base AS (
    SELECT REPLACE(REPLACE(SPLIT_PART(treeubi.ancestors::text, ',', 2), '{', ''), '"',
    '') AS resource, REPLACE(SPLIT_PART(treeubi.ancestorscode::text, ',', 2), '{', '')
    AS rescode, objiter.month, objiter.year, COUNT(safetyactions.*) AS tasks,
    kpiext.fte, COUNT(safetyactions.):FLOAT/fte AS rate FROM objiter JOIN project ON
    objiter.codice = project.codice JOIN treeubi ON project.codubi = treeubi.codice
    JOIN safetyactions ON project.codice = safetyactions.codprj JOIN kpiext ON
    objiter.month = kpiext.month AND objiter.year = kpiext.year
    WHERE tipo = 'H' AND project.sogw = 1 AND objiter.tmcre >= 1676502000 AND
    objiter.tmcre <= 1705391447 AND REPLACE(SPLIT_PART(treeubi.ancestorscode::text,
    ',', 2), '{', '') = kpiext.codubi
    GROUP BY resource, rescode, objiter.month, objiter.year, fte ORDER BY YEAR DESC,
    MONTH DESC)
    SELECT base.resource, base.rescode AS xasys, SUM(tasks) AS v_task, SUM(fte) AS
    v_fte, SUM(tasks):FLOAT/SUM(fte) AS v_rate FROM base GROUP BY resource, xasys

```

- A livello di Franke sarà la somma di tutte le actions del periodo stabilito sulla somma del numero FTE del periodo stabilito (per ogni organization Franke)

```

WITH RECURSIVE treeubi AS (
    SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.tiposet, ubicaz.codubib,
    ARRAY[]::VARCHAR[] || ubicaz.descrizio AS ancestors, ARRAY[]::VARCHAR[] ||
    ubicaz.codice AS ancestorscode, ARRAY[]::VARCHAR[] || ubicaz.tiposet AS
    ancestorsts, mancost
    FROM ubicaz WHERE codice = 'franke'
    UNION ALL
    SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.tiposet, ubicaz.codubib,
    treeubi.ancestors || ubicaz.descrizio, treeubi.ancestorscode || ubicaz.codice,
    treeubi.ancestorsts || ubicaz.tiposet, ubicaz.mancost
    FROM ubicaz, treeubi
    WHERE ubicaz.codubib = treeubi.codice
),

kpiext AS (SELECT kpiubiext.* FROM kpiubiext JOIN treeubi ON kpiubiext.codubi =
treeubi.codice),
base AS (
SELECT REPLACE(REPLACE(SPLIT_PART(treeubi.ancestors::text, ',', 2), '{', ''), '"',
') AS resource, REPLACE(SPLIT_PART(treeubi.ancestorscode::text, ',', 2), '{', ')
AS rescodes, objiter.month, objiter.year, COUNT(safetyactions.*) AS tasks,
kpiext.fte, COUNT(safetyactions.):FTE/fte AS rate FROM objiter JOIN project ON
objiter.codice = project.codice JOIN treeubi ON project.codubi = treeubi.codice
JOIN safetyactions ON project.codice = safetyactions.codprj JOIN kpiext ON
objiter.month = kpiext.month AND objiter.year = kpiext.year
WHERE tipo = 'H' AND project.sogw = 1 AND objiter.tmcre >= 1676502000 AND
objiter.tmcre <= 1705391447 AND REPLACE(SPLIT_PART(treeubi.ancestorscode::text,
',', 3), '{', ') = kpiext.codubi
GROUP BY resource, rescodes, objiter.month, objiter.year, fte ORDER BY YEAR DESC,
MONTH DESC)
SELECT base.resource, base.rescodes AS xasys, SUM(tasks) AS v_sumtask, SUM(fte) AS
v_sumfte, SUM(tasks)::FTE/SUM(fte) AS v_rate FROM base GROUP BY resource, xasys

```

Chiamata Agent KPI #5

PLANT level:

- Nome dell'agente: graph
 - ask
 - request: deviations
 - p1: rec(codubi,fromtm,totm,tiposet)
 - codubi = codice ubicaz
 - fromtm = tm di inizio
 - totm = tm di fine
 - tiposet = tipo settore (S)
 - retv: rec(resource,year,xasys,v_task,v_fte,v_rate) ESEMPIO:
 - resource = "FKT"
 - year = 2024
 - xasys = 1
 - v_task = 5
 - v_fte = 14
 - v_rate = 0.35714285714285715

ORGANIZATIONAL level:

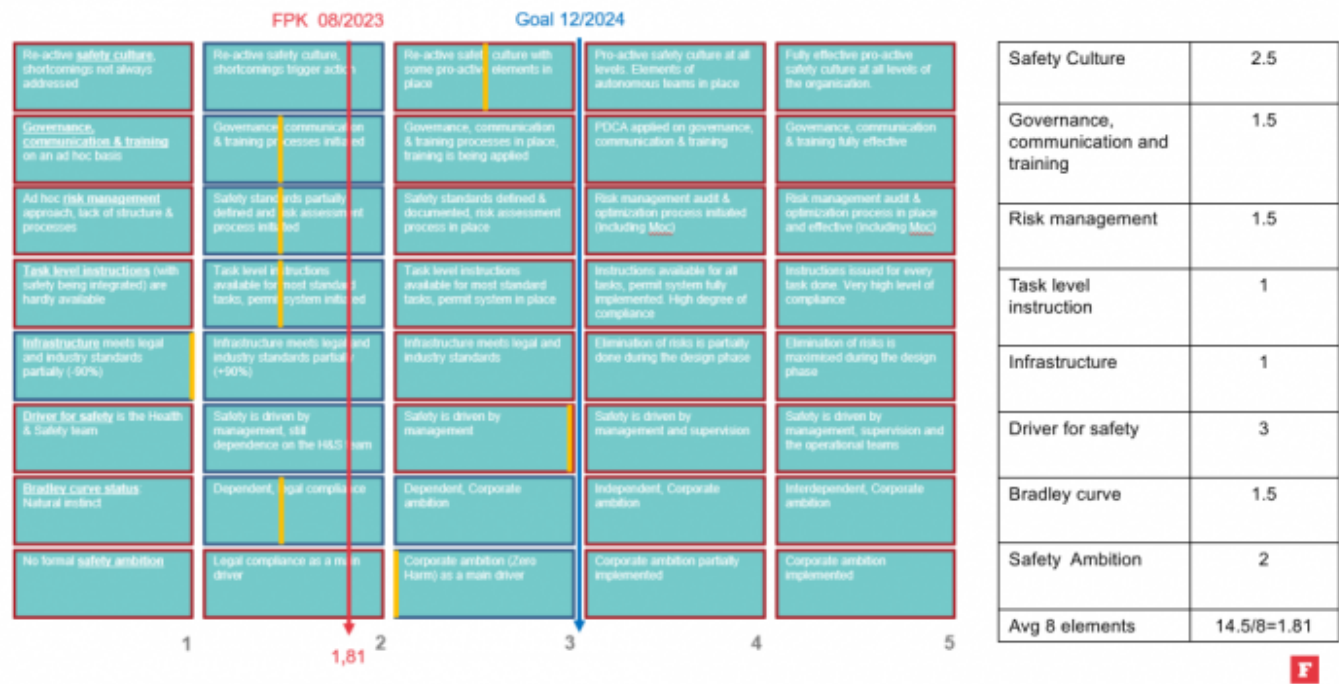
- Nome dell'agente: prestat
 - ask
 - request: kpi5.organization

- p1: rec(codubi,fromtm,totm)
 - codubi = codice ubicaz
 - fromtm = tm di inizio
 - totm = tm di fine
- retv: rec(resource,xasys,v_task,v_fte,v_rate) ESEMPIO:
 - resource = "CIC Board Plant FBS Germany Bad Säkingen"
 - xasys: "FBS"
 - v_task = 1
 - v_fte = 9
 - v_rate = 0.1111111111111111

FRANKE level:

- Nome dell'agente: prepstat
 - ask
 - request: kpi5.franke
 - p1: rec(codubi,fromtm,totm)
 - codubi = codice ubicaz
 - fromtm = tm di inizio
 - fromtm = tm di fine
 - retv: rec(resource,xasys,v_sumtask,v_sumfte,v_rate) ESEMPIO:
 - resource = "Sinks"
 - xasys= "franke001"
 - v_sumtask = 134
 - v_sumfte = 197
 - v_rate = 0.6802030456852792

KPI #6 - Maturity Assessment Score



Maturity Assessment score

3

Franke has the ambition to move to an innovative and interdependent Safety Maturity culture where Safety is fully integrated in all business processes.

Context:

The development of safety culture refers to the collective attitudes, beliefs, values, a behaviors regarding safety within an organization or society. It encompasses the way safe is perceived, prioritized, and integrated into everyday practices. Safety culture development is essential for fostering a safe and healthy environment, reducing accidents, and ensure the well-being of individuals.

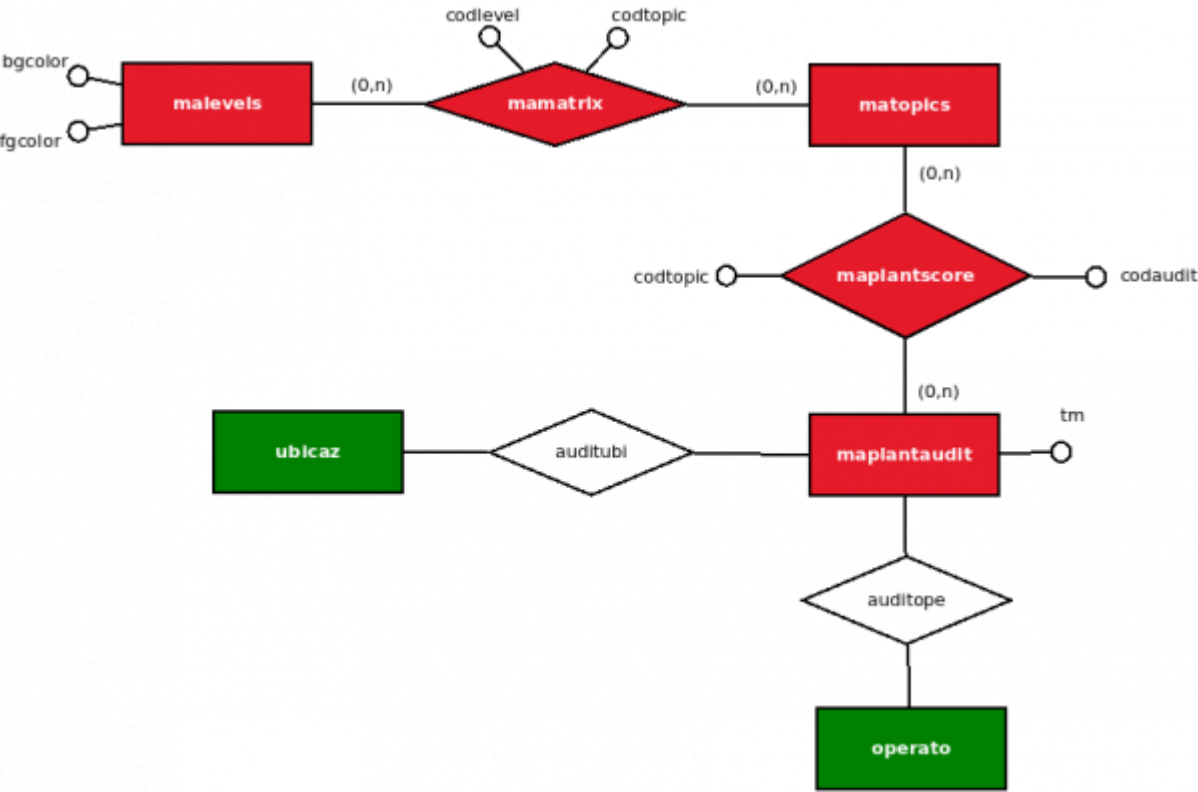
Metrics:
Score per element overall and per BU (numbers in table and can be plotted overall and per BU in spider diagram



Elements	Target 2024	Overall Average	SC						Stakes				
			Average	FP	FFC	FDS	Berie	FPW	Average	FKT	FFK	Midland	FM
Safety Culture	3	1.74	1.7	2	1	2	2	1.5	2.25	2.5	2.5	1.5	2.5
Governance & Training & Communication	3	1.98	1.9	2	2	2	2	1.5	2	2	1.5	2	2.5
Risk Management	3	1.6	1.5	2	1	1.5	2	1	1.875	2	1.5	2	2
Task Level Instructions	3	2	2	2	2	2	2	2	1.75	2	1.5	1.5	2
Infrastructure	3	1.5	1.5	1	2	1	2	1.5	1.625	2	1	2	1.5
Driver for Safety	3	1.52	1.6	2	1	1.5	2	1.5	2.25	2	3	1	3
Bradley Curve	3	2.36	2.3	3	2	2	2.5	2	2.125	2.5	1.5	2	2.5
Safety Ambition	3	2.48	2.4	3	2	2.5	2.5	2	2.375	2.5	2	2	3
Score	3.08	1.91	1.86	2.13	1.63	1.81	2.13	1.63	2.63	2.19	1.81	1.75	2.38

6

F



```
CREATE TABLE public.malevels (  
  codice INTEGER NOT NULL,  
  descrizio CHARACTER VARYING(51),  
  score INTEGER,  
  bgcolor CHARACTER VARYING(51),  
  fgcolor CHARACTER VARYING(51)  
);  
ALTER TABLE public.malevels OWNER TO postgres;  
INSERT INTO public.malevels VALUES (1, 'Pathologic', 1, 'red', '#FFFFFF');  
INSERT INTO public.malevels VALUES (2, 'Reactive', 2, 'orange', '#FFFFFF');
```



```
INSERT INTO public.malevels VALUES (3, 'Calculative', 3, 'yellow', '#000000');
INSERT INTO public.malevels VALUES (4, 'Proactive', 4, 'blue', '#FFFFFF');
INSERT INTO public.malevels VALUES (5, 'Innovative', 5, 'green', '#FFFFFF');
ALTER TABLE ONLY public.malevels ADD CONSTRAINT ix_malevels_pk PRIMARY KEY
(codice);

CREATE TABLE public.matopics (
    codice INTEGER NOT NULL,
    descrizio CHARACTER VARYING(51),
    zord INTEGER
);
ALTER TABLE public.matopics OWNER TO postgres;
INSERT INTO public.matopics VALUES (1, 'Safety Culture', 1);
INSERT INTO public.matopics VALUES (2, 'Governance, communication and training',
2);
INSERT INTO public.matopics VALUES (3, 'Risk management', 3);
INSERT INTO public.matopics VALUES (4, 'Task level instruction', 4);
INSERT INTO public.matopics VALUES (5, 'Infrastructure', 5);
INSERT INTO public.matopics VALUES (6, 'Driver for safety', 6);
INSERT INTO public.matopics VALUES (7, 'Bradley curve', 7);
INSERT INTO public.matopics VALUES (8, 'Safety Ambition', 8);
ALTER TABLE ONLY public.matopics ADD CONSTRAINT ix_matopics_pk PRIMARY KEY
(codice);

CREATE TABLE public.mamatrix (
    codice INTEGER NOT NULL,
    descrizio CHARACTER VARYING(1001),
    codlevel INTEGER,
    codtopic INTEGER
);
ALTER TABLE public.mamatrix OWNER TO postgres;
COMMENT ON COLUMN public.mamatrix.codlevel IS 'Riferimento a malevels.codice';
COMMENT ON COLUMN public.mamatrix.codtopic IS 'Riferimento a matopics.codice';

INSERT INTO public.mamatrix VALUES (1, 'Re-active safety culture, shortcomings not
always addressed', 1, 1);
INSERT INTO public.mamatrix VALUES (2, 'Re-active safety culture, shortcomings
trigger action', 2, 1);
INSERT INTO public.mamatrix VALUES (3, 'Re-active safety culture with some pro-
active elements in place', 3, 1);
INSERT INTO public.mamatrix VALUES (4, 'Pro-active safety culture at all levels.
Elements of autonomous teams in place', 4, 1);
INSERT INTO public.mamatrix VALUES (5, 'Fully effective pro-active safety culture
at all levels of the organisation', 5, 1);
INSERT INTO public.mamatrix VALUES (6, 'Governance, communication & training on an
ad hoc basis', 1, 2);
INSERT INTO public.mamatrix VALUES (7, 'Governance, communication & training
processes initiated', 2, 2);
INSERT INTO public.mamatrix VALUES (8, 'Governance, communication & training
processes in place, training is being applied', 3, 2);
INSERT INTO public.mamatrix VALUES (9, 'PDCA applied on governance, communication &
training', 4, 2);
INSERT INTO public.mamatrix VALUES (10, 'GOvernance, communication & training fully
effective', 5, 2);
```

```

INSERT INTO public.mamatrix VALUES (11, 'Ad hoc risk management approach, lack of
structure & processes', 1, 3);
INSERT INTO public.mamatrix VALUES (12, 'Safety standards partially defined and
risk assessment process initiated', 2, 3);
INSERT INTO public.mamatrix VALUES (13, 'Safety Standards defined & documented,
risk assessment process in place', 3, 3);
INSERT INTO public.mamatrix VALUES (14, 'Risk management audit & optimization
process initiated (including Moc)', 4, 3);
INSERT INTO public.mamatrix VALUES (15, 'Risk management audit & optimization
process in place and effective (including Moc)', 5, 3);
INSERT INTO public.mamatrix VALUES (16, 'Task level instructions (with safety being
integrated) are hardly available', 1, 4);
INSERT INTO public.mamatrix VALUES (17, 'Task level instructions available for most
standard tasks, permit system initiated', 2, 4);
INSERT INTO public.mamatrix VALUES (18, 'Task level instructions available for most
standard tasks, permit system in place', 3, 4);
INSERT INTO public.mamatrix VALUES (19, 'Instructions available for all tasks,
permits system fully implemented. High degree of compliance', 4, 4);
INSERT INTO public.mamatrix VALUES (20, 'Instructions issued for every task done.
Very high level of compliance', 5, 4);
INSERT INTO public.mamatrix VALUES (21, 'Infrastructure meets legal and industry
standards partially (<90%)', 1, 5);
INSERT INTO public.mamatrix VALUES (22, 'Infrastructure meets legal and industry
standards partially (>90%)', 2, 5);
INSERT INTO public.mamatrix VALUES (23, 'Infrastructure meets legal and industry
standards', 3, 5);
INSERT INTO public.mamatrix VALUES (24, 'Elimination of risks is partially done
during the design phase', 4, 5);
INSERT INTO public.mamatrix VALUES (25, 'Elimination of risks is maximised during
the design phase', 5, 5);
INSERT INTO public.mamatrix VALUES (26, 'Driver for safety is the Health & Safety
team', 1, 6);
INSERT INTO public.mamatrix VALUES (27, 'Safety is driven by management, still
dependence on the H&S team', 2, 6);
INSERT INTO public.mamatrix VALUES (28, 'Safety is driven by management', 3, 6);
INSERT INTO public.mamatrix VALUES (29, 'Safety is driven by management and
supervision', 4, 6);
INSERT INTO public.mamatrix VALUES (30, 'Safety is driven by management and
supervision and the operational teams', 5, 6);
INSERT INTO public.mamatrix VALUES (31, 'Bradley curve status: Natural instinct',
1, 7);
INSERT INTO public.mamatrix VALUES (32, 'Dependent, legal compliance', 2, 7);
INSERT INTO public.mamatrix VALUES (33, 'Dependent, Corporate Ambition', 3, 7);
INSERT INTO public.mamatrix VALUES (34, 'Independent, Corporate Ambition', 4, 7);
INSERT INTO public.mamatrix VALUES (35, 'Interdependent, Corporate Ambition', 5,
7);
INSERT INTO public.mamatrix VALUES (36, 'No formal safety ambition', 1, 8);
INSERT INTO public.mamatrix VALUES (37, 'Legal compliance as a main driver', 2, 8);
INSERT INTO public.mamatrix VALUES (38, 'Corporate ambition (Zero Harm) as a main
driver', 3, 8);
INSERT INTO public.mamatrix VALUES (39, 'Corporate ambition partially implemented',
4, 8);
INSERT INTO public.mamatrix VALUES (40, 'Corporate ambition implemented', 5, 8);

ALTER TABLE ONLY public.mamatrix ADD CONSTRAINT ix_mamatrix_pk PRIMARY KEY
(codice);

```

```

ALTER TABLE ONLY public.mamatrix ADD CONSTRAINT ix_mamatrix_codlevel FOREIGN KEY
(codlevel) REFERENCES public.malevels(codice) ON UPDATE CASCADE ON DELETE RESTRICT;
ALTER TABLE ONLY public.mamatrix ADD CONSTRAINT ix_mamatrix_codtopic FOREIGN KEY
(codtopic) REFERENCES public.matopics(codice) ON UPDATE CASCADE ON DELETE RESTRICT;

CREATE TABLE public.maplantaudit (
    codice INTEGER NOT NULL,
    codubi CHARACTER VARYING(51) NOT NULL,
    tm DOUBLE PRECISION NOT NULL,
    DATA CHARACTER VARYING(51),
    codoper CHARACTER VARYING(51)
);
ALTER TABLE public.maplantaudit OWNER TO postgres;

ALTER TABLE ONLY public.maplantaudit ADD CONSTRAINT ix_maplantaudit_pk PRIMARY KEY
(codice);
ALTER TABLE ONLY public.maplantaudit ADD CONSTRAINT ix_maplantaudit_codoper FOREIGN
KEY (codoper) REFERENCES public.operato(codice) ON UPDATE CASCADE ON DELETE
RESTRICT;
ALTER TABLE ONLY public.maplantaudit ADD CONSTRAINT ix_maplantaudit_codubi FOREIGN
KEY (codubi) REFERENCES public.ubicaz(codice) ON UPDATE CASCADE ON DELETE CASCADE;

CREATE TABLE public.maplantscore (
    codaudit INTEGER NOT NULL,
    score DOUBLE PRECISION,
    codtopic INTEGER NOT NULL
);
ALTER TABLE public.maplantscore OWNER TO postgres;
COMMENT ON COLUMN public.maplantscore.codaudit IS 'Riferimento a
maplantaudit.codice';
COMMENT ON COLUMN public.maplantscore.score IS 'Voto che viene dato nel singolo
audit, da 1.0 a 5.0';
ALTER TABLE ONLY public.maplantscore ADD CONSTRAINT ix_maplantscore_codtopic
FOREIGN KEY (codtopic) REFERENCES public.matopics(codice) ON UPDATE CASCADE ON
DELETE CASCADE;
ALTER TABLE ONLY public.maplantscore ADD CONSTRAINT ix_maplantscore_codaudit
FOREIGN KEY (codaudit) REFERENCES public.maplantaudit(codice) ON UPDATE CASCADE ON
DELETE CASCADE;

```

Ogni plant deve poter impostare il suo target per il maturity assessment. Aggiunta di un campo in ubicaz che gestisce questo target:

```

ALTER TABLE ubicaz ADD COLUMN maturitytarget DOUBLE PRECISION;

```

KPI

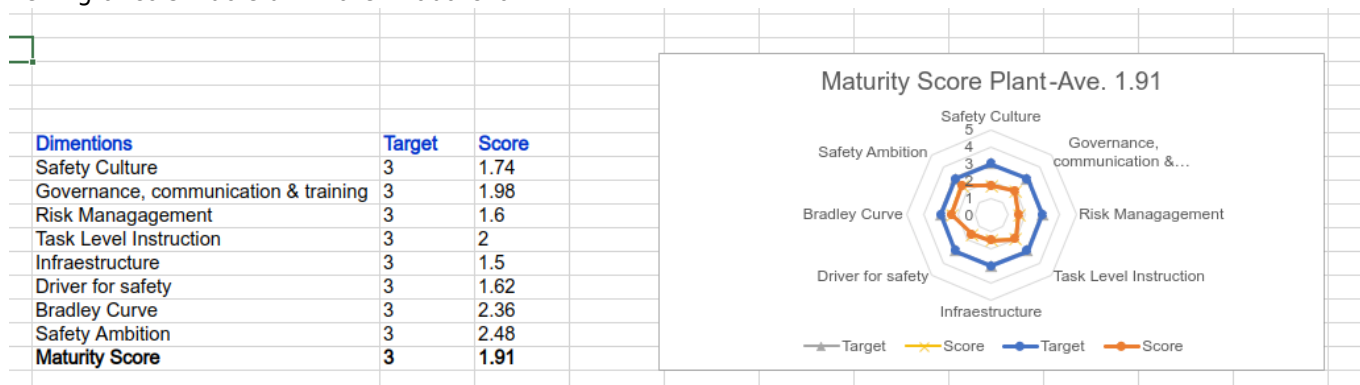
NB: Livello stabilimento ⇒ ubicaz.tiposet = 'S'. Livello di organizzazione ⇒ ubicaz.tiposet = 'A'. Franke è suddivisa in 5 organizzazioni: Sinks, Hoodss, Food Service, Supply Chain, Coffee System. Quando si parla a livello di plant si intende la singola ubicaz, quindi il codice di un plant

Quando si parla a livello di organizzazione, si intende tutte le ubicaz figlie di una certa organizzazione. Esempio

→ organizzazione sinks, tutte le ubicaz di sinks

Quando si parla a livello di franke, si intende tutte le ubicaz di tipo 'A', ovvero tutte le organizzazioni.

Per il grafico si vuole utilizzare il radarchart:



! QUERY PER I GRAFICI

- Livello di stabilimento:

Si prende l'ultimo audit inserito:

```
WITH audit AS (SELECT * FROM maplantaudit WHERE codubi = 'FKT' AND tm >= 1674169200
AND tm <= 1705311642 ORDER BY tm DESC LIMIT 1)
SELECT ubicaz.maturitytarget AS v_target, matopics.codice AS codtopic,
ubicaz.codice, ubicaz.descrizio, ubicaz.maturitytarget, matopics.descrizio AS
xasys, maplantscore.score AS v_score, tm, codaudit
FROM audit JOIN maplantscore ON audit.codice = maplantscore.codaudit JOIN matopics
ON maplantscore.codtopic = matopics.codice JOIN ubicaz ON audit.codubi =
ubicaz.codice
ORDER BY matopics.zord
```

- Organizational & Franke Level

In questo caso i radarcharts saranno 2, uno con la media voti per location e l'altro con la media voti per topic. Di seguito le due query:

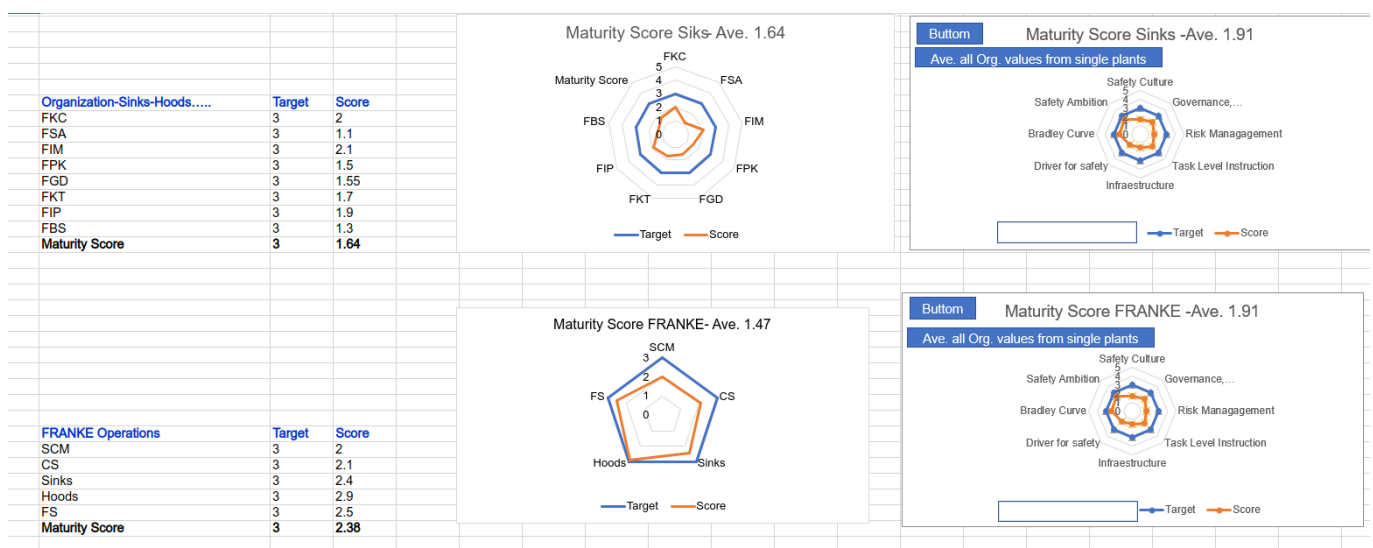
LOCATION:

```
WITH RECURSIVE treeubi AS (
SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.maturitytarget, ubicaz.tiposet,
ubicaz.codubib, ARRAY[::VARCHAR[] || ubicaz.descrizio AS ancestors,
ARRAY[::VARCHAR[] || ubicaz.codice AS ancestorscode, ARRAY[::VARCHAR[] ||
ubicaz.tiposet AS ancestorsts, mancost
FROM ubicaz WHERE codice = 'franke001'
UNION ALL
SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.maturitytarget, ubicaz.tiposet,
ubicaz.codubib, treeubi.ancestors || ubicaz.descrizio, treeubi.ancestorscode ||
ubicaz.codice, treeubi.ancestorsts || ubicaz.tiposet, ubicaz.mancost
FROM ubicaz, treeubi
WHERE ubicaz.codubib = treeubi.codice
),
lastaudit AS (SELECT MAX(tm) AS tm, codubi FROM maplantaudit JOIN treeubi ON
maplantaudit.codubi = treeubi.codice WHERE tm >= 1674169200 AND tm <= 1705311642
GROUP BY codubi),
audit AS (SELECT maplantaudit.* FROM maplantaudit JOIN lastaudit ON maplantaudit.tm
= lastaudit.tm AND maplantaudit.codubi = lastaudit.codubi)
SELECT REPLACE(REPLACE(REPLACE(SPLIT_PART(treeubi.ancestors::text, ',', 2),
```

```
{','), '}', ' '), ' ', ' ') AS resource,
REPLACE(REPLACE(SPLIT_PART(treeubi.ancestorscode::text, ',', 2), '{', ' '), '}', ' ')
AS xasys, treeubi.maturitytarget AS v_target, AVG(maplantscore.score) AS v_score
FROM audit JOIN maplantscore ON audit.codice = maplantscore.codaudit JOIN matopics
ON maplantscore.codtopic = matopics.codice JOIN treeubi ON audit.codubi =
treeubi.codice GROUP BY resource, xasys, maturitytarget
```

TOPIC:

```
WITH RECURSIVE treeubi AS (
    SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.maturitytarget, ubicaz.tiposet,
    ubicaz.codubib, ARRAY[]::VARCHAR[] || ubicaz.descrizio AS ancestors,
    ARRAY[]::VARCHAR[] || ubicaz.codice AS ancestorscode, ARRAY[]::VARCHAR[] ||
    ubicaz.tiposet AS ancestorsts, mancost
    FROM ubicaz WHERE codice = 'franke001'
    UNION ALL
    SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.maturitytarget, ubicaz.tiposet,
    ubicaz.codubib, treeubi.ancestors || ubicaz.descrizio, treeubi.ancestorscode ||
    ubicaz.codice, treeubi.ancestorsts || ubicaz.tiposet, ubicaz.mancost
    FROM ubicaz, treeubi
    WHERE ubicaz.codubib = treeubi.codice
),
lastaudit AS (SELECT MAX(tm) AS tm, codubi FROM maplantaudit JOIN treeubi ON
maplantaudit.codubi = treeubi.codice WHERE tm >= 1674169200 AND tm <= 1705311642
GROUP BY codubi),
audit AS (SELECT maplantaudit.* FROM maplantaudit JOIN lastaudit ON maplantaudit.tm
= lastaudit.tm AND maplantaudit.codubi = lastaudit.codubi)
SELECT matopics.codice AS codtopic, matopics.descrizio AS xasys,
treeubi.maturitytarget AS v_target, AVG(maplantscore.score) AS v_score FROM audit
JOIN maplantscore ON audit.codice = maplantscore.codaudit JOIN matopics ON
maplantscore.codtopic = matopics.codice JOIN treeubi ON audit.codubi =
treeubi.codice GROUP BY matopics.codice, matopics.descrizio, maturitytarget
```



La media dei punteggi dell'ultimo audit per ogni ubicaz dell'organizzazione (nel caso di franke, per ogni organizzazione di franke)

Chiamata Agent KPI #6

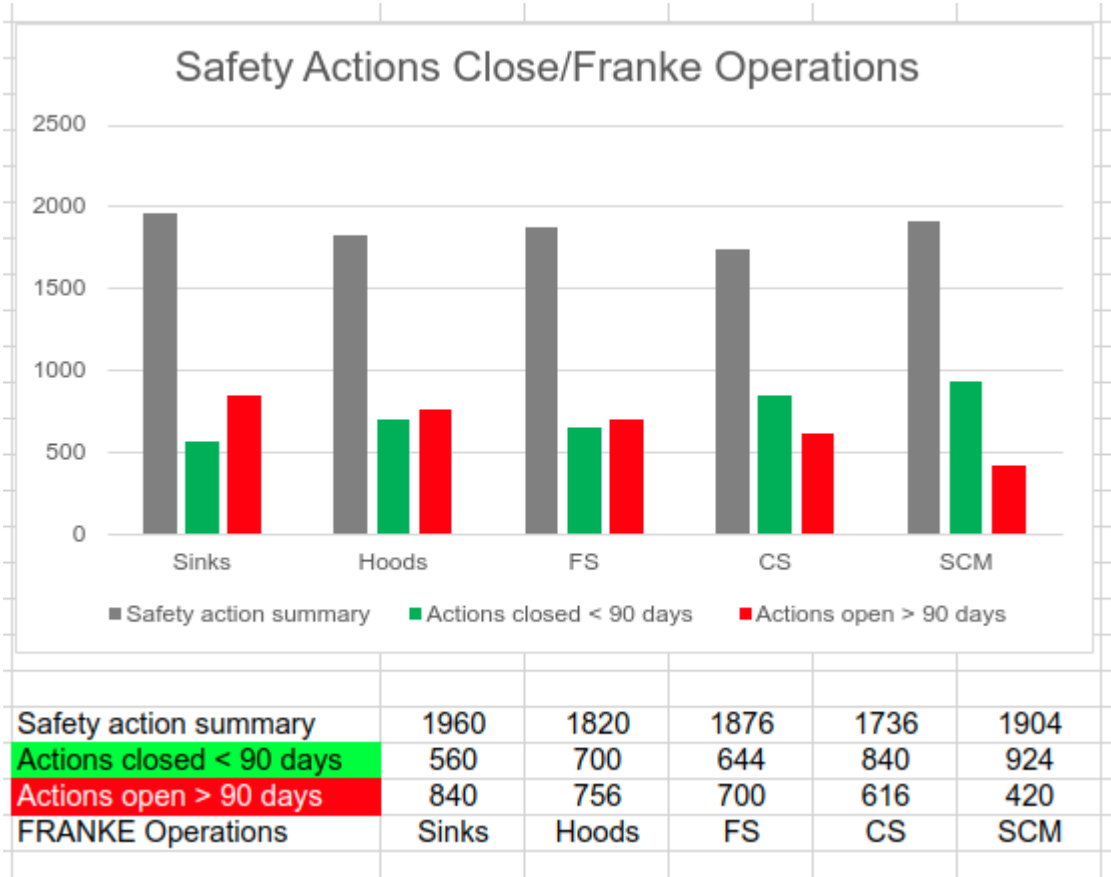
PLANT level:

- Nome dell'agente: prestat
 - ask
 - request: kpi6.plant
 - p1: rec(codubi,fromtm,totm)
 - codubi = codice ubicaz
 - fromtm = tm di inizio
 - totm = tm di fine
 - retv: rec(v_target,codtopic,codice,descrizio,maturitytarget,xasys,v_score,tm,codaudit)
 - ESEMPIO:
 - v_target = 3
 - codtopic = 1
 - codice = "FKT"
 - descrizio = "CIC Board Plant FKT Switzerland Aarburg"
 - maturitytarget = 3
 - xasys = "Safety Culture"
 - v_score = 2.8
 - tm = 1705327488
 - codaudit = 12

ORGANIZATIONAL and FRANKE level:

- Nome dell'agente: prestat
 - ask
 - request: kpi6
 - p1: rec(_subkey,codubi,fromtm,totm)
 - _subkey = subkey delle query da eseguire divise da virgola (location,topic)
 - codubi = codice ubicaz
 - fromtm = tm di inizio
 - totm = tm di fine
 - retv: **i valori di "kpi6.location" e "kpi6.topic" ritorneranno come stringhe!**

```
[
  {
    "kpi6.location": [
      {
        "resource": "CIC Plant FIM IND Aurangabad",
        "xasys": "FIM",
        "v_target": 3.0,
        "v_score": 2.3999999999999999
      },
      ...
    ],
    "kpi6.topic": [
      {
        "codtopic": 2,
        "xasys": "Governance, communication and training",
        "v_target": 3.0,
        "v_score": 2.3999999999999999
      },
      ...
    ]
  }
]
```

Esempio di due stack diversi sullo stesso chart

```

33  {,
34
35  plotOptions: {
36    column: {
37      stacking: 'normal'
38    }
39  },
40
41  series: [{
42    name: 'Norway',
43    data: [148, 133, 124],
44    stack: 'Europe'
45  }, {
46    name: 'Germany',
47    data: [102, 98, 65],
48    stack: 'Europe'
49  }, {
50    name: 'United States',
51    data: [113, 122, 95],
52    stack: 'North America'
53  }, {
54    name: 'Canada',
55    data: [77, 72, 80],
56    stack: 'North America'
57  }]
58  });

```

Olympic Games all-time medal table, grouped by continent

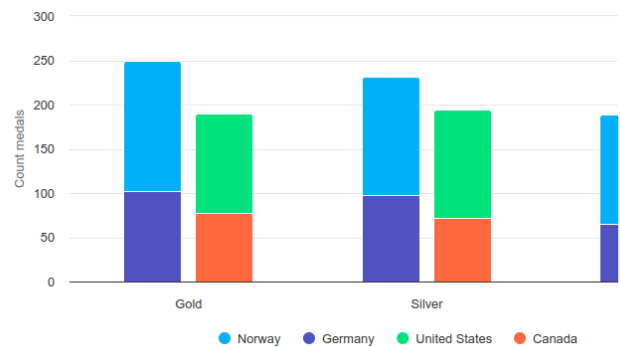


Chart showing stacked columns with grouping, allowing specific series to be stacked on the same column. Stacking is often used to visualize data that accumulates to a sum.

! QUERY PER I GRAFICI

```

WITH RECURSIVE treeubi AS (SELECT u.* FROM ubicaz u WHERE u.codice = 'FKT' UNION
ALL SELECT u.* FROM ubicaz u, treeubi WHERE u.codubib = treeubi.codice),
allacts AS (SELECT EXTRACT(MONTH FROM TIMESTAMP WITH TIME ZONE 'epoch' + tmcre *
INTERVAL '1 second') AS MONTH, EXTRACT(YEAR FROM TIMESTAMP WITH TIME ZONE 'epoch' +
tmcre * INTERVAL '1 second') AS YEAR, COUNT(safetyactions.*) AS total FROM project
JOIN treeubi ON project.codubi = treeubi.codice JOIN safetyactions ON
project.codice = safetyactions.codprj WHERE safetyactions.tmcre >= 1676502000 AND

```

```

safetyactions.tmcre <= 1705391447 GROUP BY MONTH, YEAR),
closedacts AS (SELECT EXTRACT(MONTH FROM TIMESTAMP WITH TIME ZONE 'epoch' + tmcre *
INTERVAL '1 second') AS MONTH, EXTRACT(YEAR FROM TIMESTAMP WITH TIME ZONE 'epoch' +
tmcre * INTERVAL '1 second') AS YEAR, COUNT(safetyactions.*) AS numclosed FROM
project JOIN treeubi ON project.codubi = treeubi.codice JOIN safetyactions ON
project.codice = safetyactions.codprj WHERE safetyactions.tmcre >= 1676502000 AND
safetyactions.tmcre <= 1705391447 AND safetyactions.status = 'C' AND
(safetyactions.tmclosed-safetyactions.tmcre)/86400 < 90 GROUP BY MONTH, YEAR),
openacts AS (SELECT EXTRACT(MONTH FROM TIMESTAMP WITH TIME ZONE 'epoch' + tmcre *
INTERVAL '1 second') AS MONTH, EXTRACT(YEAR FROM TIMESTAMP WITH TIME ZONE 'epoch' +
tmcre * INTERVAL '1 second') AS YEAR, COUNT(safetyactions.*) AS numopen FROM
project JOIN treeubi ON project.codubi = treeubi.codice JOIN safetyactions ON
project.codice = safetyactions.codprj WHERE safetyactions.tmcre >= 1676502000 AND
safetyactions.tmcre <= 1705391447 AND safetyactions.status = 'O' AND (tm()-
safetyactions.tmcre)/86400 > 90 GROUP BY MONTH, YEAR)

SELECT DISTINCT 'FKT' AS resource, allacts.month AS xasys, allacts.year, total AS
v_total, numclosed AS v_numclosed, numopen AS v_numopen
FROM allacts LEFT JOIN closedacts ON allacts.month = closedacts.month AND
allacts.year = closedacts.year LEFT JOIN openacts ON allacts.month = openacts.month
AND allacts.year = openacts.year ORDER BY YEAR DESC, xasys DESC

```

resource	month	year	numclosed	numopen	total
FKT	1	2024	NULL	NULL	8
FKT	10	2023	1	NULL	2
FKT	9	2023	17	20	40
FKT	8	2023	22	1	23
FKT	7	2023	19	NULL	19
FKT	6	2023	107	6	119
FKT	5	2023	50	6	58
FKT	4	2023	41	NULL	47
FKT	3	2023	53	1	81
FKT	2	2023	14	NULL	26

Organization Level:

```

WITH RECURSIVE treeubi AS (
    SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.tiposet, ubicaz.codubib,
    ARRAY[]::VARCHAR[] || ubicaz.descrizio AS ancestors, ARRAY[]::VARCHAR[] ||
    ubicaz.codice AS ancestorscode, ARRAY[]::VARCHAR[] || ubicaz.tiposet AS
    ancestorsts, mancost
    FROM ubicaz WHERE codice = 'franke001'
    UNION ALL
    SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.tiposet, ubicaz.codubib,
    treeubi.ancestors || ubicaz.descrizio, treeubi.ancestorscode || ubicaz.codice,
    treeubi.ancestorsts || ubicaz.tiposet, ubicaz.mancost
    FROM ubicaz, treeubi
    WHERE ubicaz.codubib = treeubi.codice
),
allacts AS (SELECT REPLACE(REPLACE(REPLACE(SPLIT_PART(ancestors::text, ',', 2),
'{', ''), '}', ''), '"', '') AS resource,
REPLACE(REPLACE(SPLIT_PART(ancestorscode::text, ',', 2), '{', ''), '}', '') AS
rescode, COUNT(safetyactions.*) AS total FROM project JOIN treeubi ON
project.codubi = treeubi.codice JOIN safetyactions ON project.codice =
safetyactions.codprj WHERE safetyactions.tmcre >= 1676502000 AND
safetyactions.tmcre <= 1705391447 GROUP BY resource, rescode),

```

```
closedacts AS (SELECT REPLACE(REPLACE(REPLACE(SPLIT_PART(ancestors::text, ',', 2),
'{', ''), '}', ''), '', '') AS resource,
REPLACE(REPLACE(SPLIT_PART(ancestorscode::text, ',', 2), '{', ''), '}', '') AS
rescode, COUNT(safetyactions.*) AS numclosed FROM project JOIN treeubi ON
project.codubi = treeubi.codice JOIN safetyactions ON project.codice =
safetyactions.codprj WHERE safetyactions.tmcre >= 1676502000 AND
safetyactions.tmcre <= 1705391447 AND safetyactions.status = 'C' AND
(safetyactions.tmclosed-safetyactions.tmcre)/86400 < 90 GROUP BY resource,
rescode),
```

```
openacts AS (SELECT REPLACE(REPLACE(REPLACE(SPLIT_PART(ancestors::text, ',', 2),
'{', ''), '}', ''), '', '') AS resource,
REPLACE(REPLACE(SPLIT_PART(ancestorscode::text, ',', 2), '{', ''), '}', '') AS
rescode, COUNT(safetyactions.*) AS numopen FROM project JOIN treeubi ON
project.codubi = treeubi.codice JOIN safetyactions ON project.codice =
safetyactions.codprj WHERE safetyactions.tmcre >= 1676502000 AND
safetyactions.tmcre <= 1705391447 AND safetyactions.status = 'O' AND (tm()-
safetyactions.tmcre)/86400 > 90 GROUP BY resource, rescode)
```

```
SELECT allacts.resource, allacts.rescode AS xasys, total AS v_total, numclosed AS
v_numclosed, numopen AS v_numopen FROM allacts LEFT JOIN closedacts ON
allacts.rescode = closedacts.rescode LEFT JOIN openacts ON allacts.rescode =
openacts.rescode ORDER BY resource
```

resource	rescode	total	numclosed	numopen
CIC Board Plant FBS Germany Bad Säckingen	FBS	1	NULL	1
CIC Board Plant FGD Slovakia Strecno	FGD	698	656	15
CIC Board Plant FKT Switzerland Aarburg	FKT	423	324	34
CIC Plant FIM IND Aurangabad	FIM	60	56	4
CIC Plant FIP Italy Peschiera del Garda	FIP	106	85	21
CIC Plant FKC Canada Midland	FKC	391	379	6
CIC Plant FPK CN Heshan	FPK	274	249	15
CIC Plant FPW Poland Sekocin Nowy	FPW	12	12	NULL
CIC Plant FSA South Africa Durban	FSA	392	297	55

Franke Level:

```
WITH RECURSIVE treeubi AS (
  SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.tiposet, ubicaz.codubib,
  ARRAY[]::VARCHAR[] || ubicaz.descrizio AS ancestors, ARRAY[]::VARCHAR[] ||
  ubicaz.codice AS ancestorscode, ARRAY[]::VARCHAR[] || ubicaz.tiposet AS
  ancestorsts, mancost
  FROM ubicaz WHERE codice = 'franke'
  UNION ALL
  SELECT ubicaz.codice, ubicaz.descrizio, ubicaz.tiposet, ubicaz.codubib,
  treeubi.ancestors || ubicaz.descrizio, treeubi.ancestorscode || ubicaz.codice,
  treeubi.ancestorsts || ubicaz.tiposet, ubicaz.mancost
  FROM ubicaz, treeubi
  WHERE ubicaz.codubib = treeubi.codice
),
allacts AS (SELECT REPLACE(REPLACE(REPLACE(SPLIT_PART(ancestors::text, ',', 2),
'{', ''), '}', ''), '', '') AS resource,
REPLACE(REPLACE(SPLIT_PART(ancestorscode::text, ',', 2), '{', ''), '}', '') AS
rescode, COUNT(safetyactions.*) AS total FROM project JOIN treeubi ON
project.codubi = treeubi.codice JOIN safetyactions ON project.codice =
```

```

safetyactions.codprj WHERE safetyactions.tmcre >= 1676502000 AND
safetyactions.tmcre <= 1705391447 GROUP BY resource, rescode),
closedacts AS (SELECT REPLACE(REPLACE(REPLACE(SPLIT_PART(ancestors::text, ',', 2),
'{' , '' ), '}' , '' ), '"' , '' ) AS resource,
REPLACE(REPLACE(SPLIT_PART(ancestorscode::text, ',', 2), '{' , '' ), '}' , '' ) AS
rescode, COUNT(safetyactions.*) AS numclosed FROM project JOIN treeubi ON
project.codubi = treeubi.codice JOIN safetyactions ON project.codice =
safetyactions.codprj WHERE safetyactions.tmcre >= 1676502000 AND
safetyactions.tmcre <= 1705391447 AND safetyactions.status = 'C' AND
(safetyactions.tmclosed-safetyactions.tmcre)/86400 < 90 GROUP BY resource,
rescode),

openacts AS (SELECT REPLACE(REPLACE(REPLACE(SPLIT_PART(ancestors::text, ',', 2),
'{' , '' ), '}' , '' ), '"' , '' ) AS resource,
REPLACE(REPLACE(SPLIT_PART(ancestorscode::text, ',', 2), '{' , '' ), '}' , '' ) AS
rescode, COUNT(safetyactions.*) AS numopen FROM project JOIN treeubi ON
project.codubi = treeubi.codice JOIN safetyactions ON project.codice =
safetyactions.codprj WHERE safetyactions.tmcre >= 1676502000 AND
safetyactions.tmcre <= 1705391447 AND safetyactions.status = 'O' AND (tm()-
safetyactions.tmcre)/86400 > 90 GROUP BY resource, rescode)

SELECT allacts.resource AS xasys, allacts.rescode, total AS v_total, numclosed AS
v_numclosed, numopen AS v_numopen FROM allacts LEFT JOIN closedacts ON
allacts.rescode = closedacts.rescode LEFT JOIN openacts ON allacts.rescode =
openacts.rescode ORDER BY xasys

```

resource	rescode	total	numclosed	numopen
Food Services	franke005	936	369	441
Franke Supply Chain	franke003	1123	1023	79
FS American Organization	0000002996	6	NULL	6
Hoods	franke004	820	505	228
Sinks	franke001	2357	2058	151
Sites Old Management	franke007	17	5	11

Chiamata Agent KPI #7

PLANT level:

- Nome dell'agente: graph
 - ask
 - request: maturity
 - p1: rec(codubi,fromtm,totm,tiposet)
 - codubi = codice ubicaz
 - fromtm = tm di inizio
 - totm = tm di fine
 - tiposet = tipologia settore (S)
 - retv: rec(resource,xasys,year,v_total,v_numclosed,v_numopen) ESEMPIO:
 - resource = "FKT"
 - xasys = 1
 - year = 2024
 - v_total = 9
 - v_numclosed = 0
 - v_numopen = 0

ORGANIZATIONAL level:

- Nome dell'agente: prepmat

- ask
 - request: kpi7.organization
 - p1: rec(codubi,fromtm,totm)
 - codubi = codice ubicaz
 - fromtm = tm di inizio
 - totm = tm di fine
 - retv: rec(resource,xasys,v_total,v_numclosed,v_numopen) ESEMPIO:
 - resource = "CIC Board Plant FBS Germany Bad Säckingen"
 - xasys = ""FBS""
 - v_total = 1
 - v_numclosed = "" (se NULL prepstat ritorna "")
 - v_numopen = 1

FRANKE level:

- Nome dell'agente: prepstat
 - ask
 - request: kpi7.franke
 - p1: rec(codubi,fromtm,totm)
 - codubi = codice ubicaz
 - fromtm = tm di inizio
 - totm = tm di fine
 - retv: rec(resource,xasys,v_total,v_numclosed,v_numopen) ESEMPIO:
 - xasys = "Food Services"
 - rescode = "franke005"
 - v_total = 936
 - v_numclosed = 369
 - v_numopen = 508

RIKS ASSESSMENT

Utilizzo di un nuovo objiter, di tipo 'U'.

Un Risk Assessment è simile, a livello di struttura, ad un JSO, per cui si andranno a modificare le pagine di gestione del JSO per poter gestire anche questo nuovo tipo di oggetto.

Per il risk assessment abbiamo bisogno di alcune tabelle anagrafiche.

```
CREATE TABLE public.rafrequency (
    codice CHARACTER VARYING(11) NOT NULL,
    descrizio CHARACTER VARYING(51),
    severity INTEGER
);
ALTER TABLE public.rafrequency OWNER TO postgres;
COMMENT ON TABLE public.rafrequency IS 'Tabella delle frequenze ammesse per il risk assessment';
INSERT INTO public.rafrequency VALUES ('C', 'Continua', 1);
INSERT INTO public.rafrequency VALUES ('D', 'Giornaliera', 2);
INSERT INTO public.rafrequency VALUES ('W', 'Settimanale', 3);
INSERT INTO public.rafrequency VALUES ('M', 'Mensile', 4);
INSERT INTO public.rafrequency VALUES ('Y', 'Annuale', 5);
ALTER TABLE ONLY public.rafrequency ADD CONSTRAINT ix_rafrequency_pk PRIMARY KEY (codice);

-- Per gestire i due nuovi campi hazard description e hazard control
```

```
ALTER TABLE project ADD COLUMN hazard VARCHAR(501);
ALTER TABLE project ADD COLUMN hazardctrl VARCHAR(501);
ALTER TABLE project ADD COLUMN rafreq VARCHAR(11);
ALTER TABLE project ADD COLUMN crlsev INTEGER;
ALTER TABLE project ADD COLUMN crlprob INTEGER;
ALTER TABLE project ADD COLUMN frlsev INTEGER;
ALTER TABLE project ADD COLUMN frlprob INTEGER;

--Severity
CREATE TABLE public.raseverity (
    codice INTEGER NOT NULL,
    descrizio CHARACTER VARYING(51),
    rating INTEGER
);
ALTER TABLE public.raseverity OWNER TO postgres;
COMMENT ON TABLE public.raseverity IS 'Tabella delle severità per il risk management';
INSERT INTO public.raseverity VALUES (1, 'Fatale o Fatale Multiplo', 16);
INSERT INTO public.raseverity VALUES (2, 'Disabilità Permanente', 8);
INSERT INTO public.raseverity VALUES (3, 'Tempo Perso', 4);
INSERT INTO public.raseverity VALUES (4, 'Cure Mediche', 2);
INSERT INTO public.raseverity VALUES (5, 'Primo Soccorso', 1);
ALTER TABLE ONLY public.raseverity ADD CONSTRAINT ix_raseverity_pk PRIMARY KEY (codice);

--Probability
CREATE TABLE public.raprobability (
    codice INTEGER NOT NULL,
    descrizio CHARACTER VARYING(51),
    rating INTEGER
);
ALTER TABLE public.raprobability OWNER TO postgres;
COMMENT ON TABLE public.raprobability IS 'Tabella delle probabilità previste nel Rik Assessment';
INSERT INTO public.raprobability VALUES (1, 'Impossibile', 1);
INSERT INTO public.raprobability VALUES (2, 'Molto Improbabile', 2);
INSERT INTO public.raprobability VALUES (3, 'Improbabile', 4);
INSERT INTO public.raprobability VALUES (4, 'Possibile', 8);
INSERT INTO public.raprobability VALUES (5, 'Probabile', 30);
INSERT INTO public.raprobability VALUES (6, 'Molto Probabile', 60);
ALTER TABLE ONLY public.raprobability ADD CONSTRAINT ix_raprobability_pk PRIMARY KEY (codice);
```

COGNOS KPI

Creo un nuovo ruolo all'interno del digcic.

Utente: cognos_user

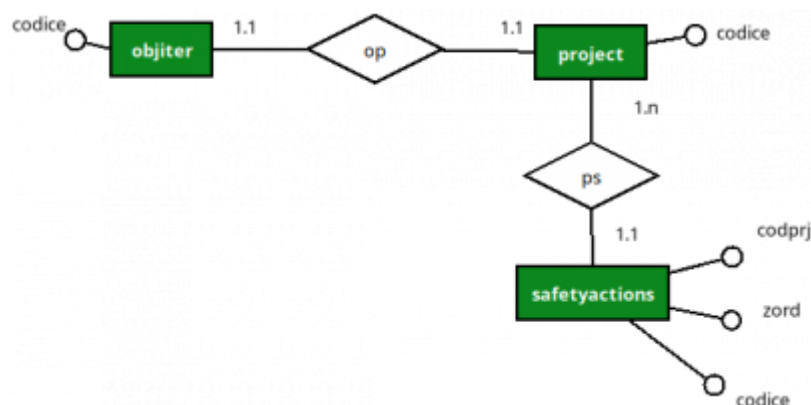
Password: W4rJA5xmC92yZ\$RGKb#iKA

Questo utente ha accesso ad una sola tabella del database, la cognoskpi (al momento, 06-03-2024, la struttura è temporanea). Una volta al mese cognos si conatterà a questa tabella per inserire i dati dei KPI.

Email Notification System

Si richiede di sviluppare un sistema di notifica tramite email quando:

- 1) Viene assegnato/riassegnato un responsabile ad una safety action
- 2) Viene assegnato/riassegnato un problem solver ad un action plan



Safety Action

Una safety action è un'azione solitamente collegata ai Job Safety Observation e alle Vulnerability Management. Ogni azione ha un responsabile che si dovrà occupare di eseguirla. Il responsabile viene identificato nel campo safetyactions.codoper

Action Plan

Esistono 4 tipologie di actionplan all'interno del DSFM. Quality, Delivery, Cost e Maintenance. Gli action plan sono objiter di tipo Q (quality), O (cost), W (delivery), I (maintenance). L'utente a cui è assegnato un actionplan si trova nel campo codoperres della tabella project

Il sistema di notifica dovrà essere configurabile ed in particolare:

- potrà essere abilitato per alcuni plant (ubicaz.tiposet = 'S').
- potrà essere abilitato per le safety action
- potrà essere abilitato per ogni singolo action plan

Ogni action plan ed ogni safety action saranno collegati ad una ubicaz (che potrà essere di tiposet S, R o P).

Se gerarchicamente l'ubicaz associata ad una safety action o ad un action plan appartiene ai plant abilitati e se l'oggetto stesso (safety action o lo specifico action plan) è abilitato all'invio della notifica, sarà possibile l'invio delle email verso il responsabile definito.

Quando inviare l'email?

- gerarchicamente l'ubicaz associata ad una safety action o ad un action plan appartiene ai plant abilitati

```

WITH RECURSIVE treeubi AS (
    SELECT u.*
    FROM ubicaz u
    WHERE u.codice = project.codubi
    UNION ALL
    SELECT u.*
    FROM ubicaz u, treeubi
    WHERE u.codice = treeubi.codubib
)
SELECT DISTINCT treeubi.*
FROM treeubi
  
```



```
WHERE tiposet = 'S'
```

- l'oggetto stesso (safety action o lo specifico action plan) è abilitato
- il responsabile dell'azione è definito

Se tutte queste condizioni sono vere, l'email dovrà essere inviata al responsabile.

Quando viene assegnato un nuovo responsabile (e rimangono valide le condizioni precedenti), il vecchio responsabile dovrà essere notificato del cambio owner ed il nuovo responsabile dovrà essere notificato dell'avvenuta assegnazione di una SA o di un AP

Creare un campo codice per la tabella safetyactions e valorizzarlo con un id incrementale:

```
ALTER TABLE safetyactions ADD COLUMN codice FLOAT;  
WITH numerati AS (  
    SELECT ctid, ROW_NUMBER() OVER () AS nuovo_codice  
    FROM safetyactions  
)  
UPDATE safetyactions  
SET codice = numerati.nuovo_codice  
FROM numerati  
WHERE safetyactions.ctid = numerati.ctid;
```