

## SQL TABELLE

```
CREATE TABLE AMMINISTRAOTRE  
(  
  nome VARCHAR2(20) NOT NULL,  
  cognome VARCHAR2(20) NOT NULL,  
  cf VARCHAR2 (16) PRIMARY KEY,  
  utente VARCHAR2(20) unique NOT NULL,  
  password VARCHAR2(20) NOT NULL  
);
```

```
CREATE TABLE FORNITORE(  
  nome VARCHAR2(20) NOT NULL,  
  cognome VARCHAR2(20) NOT NULL,  
  cf VARCHAR2(16) PRIMARY KEY  
);
```

```
CREATE TABLE OPERATORE(  
  nome VARCHAR2(20) NOT NULL,  
  cognome VARCHAR2(20) NOT NULL ,  
  cf VARCHAR2(16) PRIMARY KEY,  
  stipendio FLOAT NOT NULL,  
);
```

```
CREATE TABLE CANTIERE  
(  
  nome VARCHAR2(20) NOT NULL,  
  indirizzo VARCHAR2(40) NOT NULL,  
  comune VARCHAR2(30) NOT NULL,  
  data_inizio DATE NOT NULL,  
  data_fine DATE NOT NULL,  
  amministratore VARCHAR2(16),
```

```
cod_cantiere VARCHAR2(16) PRIMARY KEY,  
);
```

```
ALTER TABLE CANTIERE
```

```
ADD CONSTRAINT FK_CANTIERE_AMMINISTRATORE
```

```
FOREIGN KEY(amministratore) REFERENCES AMMINISTRATORE(cf) ON DELETE CASCADE;
```

```
CREATE TABLE MACCHINE
```

```
(
```

```
tipo_macchina VARCHAR2(20) CHECK (tipo_macchina='ESCAVATORE' or tipo_macchina='GRU' or  
tipo_macchina='ELEVATORI' or tipo_macchina='DUMPER' or tipo_macchina='PIATTAFORME AEREE'),
```

```
cod_macchina VARCHAR2(16) PRIMARY KEY,
```

```
cantiere VARCHAR2(16) NOT NULL
```

```
);
```

```
ALTER TABLE MACCHINE
```

```
ADD CONSTRAINT FK_CANTIERE_MACCHINE
```

```
FOREIGN KEY (cantiere) REFERENCES CANTIERE(cod_cantiere) ON DELETE CASCADE;
```

```
CREATE TABLE CAPOCANTIERE
```

```
(
```

```
nome VARCHAR2(20) NOT NULL,
```

```
cognome VARCHAR2(20) NOT NULL,
```

```
cf VARCHAR2(20) NOT NULL PRIMARY KEY,
```

```
stipendio FLOAT NOT NULL,
```

```
data_assunzione DATE NOT NULL,
```

```
cantiere VARCHAR2(16) NOT NULL,
```

```
utente VARCHAR2(20) NOT NULL UNIQUE,
```

```
password VARCHAR2(20) NOT NULL
```

```
);
```

```
ALTER TABLE CAPOCANTIERE
```

```
ADD CONSTRAINT FK_CAPOCANTIERE_CANTIERE  
FOREIGN KEY(cantiere) REFERENCES CANTIERE(cod_cantiere);
```

```
CREATE TABLE MATERIALI(  
tipo_materiale VARCHAR2(20) CHECK (tipo_materiale='LEGNO' OR tipo_materiale='ACCIAIO' OR  
tipo_materiale='VETRO' OR tipo_materiale='CERAMICHE' OR tipo_materiale='CEMENTO'),  
costo FLOAT NOT NULL,  
data_acquisto DATE NOT NULL,  
quantità INTEGER,  
cod_materiale VARCHAR2(16) PRIMARY KEY,  
cantiere VARCHAR2(16) NOT NULL,  
fornitore VARCHAR2(16) NOT NULL  
);
```

```
ALTER TABLE MATERIALI  
ADD CONSTRAINT FK_MATERIALI_CANTIERE  
FOREIGN KEY(cantiere) REFERENCES CANTIERE(cod_cantiere) ON DELETE CASCADE;
```

```
ALTER TABLE MATERIALI  
ADD CONSTRAINT FK_MATERIALI_FORNITORE  
FOREIGN KEY(fornitore) REFERENCES FORNITORE(cf) ON DELETE CASCADE;
```

```
CREATE TABLE OPERAIO  
(  
nome VARCHAR2(20) NOT NULL,  
cognome VARCHAR2(20) NOT NULL ,  
cf VARCHAR2(16) PRIMARY KEY,  
stipendio FLOAT NOT NULL,  
data_assunzione DATE NOT NULL,  
cantiere VARCHAR2(16) NOT NULL,  
macchina VARCHAR2(16),  
livello_ruolo INT CHECK(livello_ruolo>=1 and livello_ruolo<=7) NOT NULL
```

);

ALTER TABLE OPERAIO

ADD CONSTRAINT FK\_OPERAIO\_MACCHINA

FOREIGN KEY(macchina) REFERENCES Macchine(cod\_macchina);

ALTER TABLE OPERAIO

ADD CONSTRAINT FK\_OPERAIO\_CANTIERE

FOREIGN KEY(cantiere) REFERENCES CANTIERE(cod\_cantiere);

CREATE TABLE AREA\_CANTIERE

(

tipo\_area VARCHAR2(20) CHECK (tipo\_area='VERDE PUBBLICO' or tipo\_area='ZONA RESIDENZIALE' or tipo\_area='SERVIZI PUBBLICI'),

cod\_area VARCHAR2(16) PRIMARY KEY,

cantiere VARCHAR2(16) NOT NULL,

capo\_operaio VARCHAR2(16) NOT NULL

);

ALTER TABLE AREA\_CANTIERE

ADD CONSTRAINT FK\_AREA\_OPERAIO

FOREIGN KEY(capo\_operaio) REFERENCES OPERAIO(cf);

ALTER TABLE AREA\_CANTIERE

ADD CONSTRAINT FK\_AREA\_CANTIERE

FOREIGN KEY(cantiere) REFERENCES CANTIERE(cod\_cantiere) ON DELETE CASCADE;

CREATE TABLE SENSORI

(

tipo\_sensore VARCHAR2(10) CHECK(tipo\_sensore='RUMORE' or tipo\_sensore='FUGA GAS'),

valore FLOAT NOT NULL,

cod\_sensore VARCHAR2(16) PRIMARY KEY,

```
area_cantiere VARCHAR2(16) NOT NULL,  
operatore VARCHAR2(16) NOT NULL,  
soglia_allarme FLOAT CHECK(soglia_allarme>0) NOT NULL  
);
```

```
ALTER TABLE SENSORI  
ADD CONSTRAINT FK_SENSORI_OPERATORE  
FOREIGN KEY(operatore) REFERENCES OPERATORE(cf) ON DELETE CASCADE;
```

```
ALTER TABLE SENSORI  
ADD CONSTRAINT FK_SENSORI_AREA  
FOREIGN KEY(area_cantiere) REFERENCES AREA_CANTIERE(cod_area) ON DELETE CASCADE;
```

```
alter table CAPOCANTIERE  
add constraint STIPENDIO_CAPOCANTIERE check(STIPENDIO>0) ENABLE;
```

```
alter table OPERATORE  
add constraint STIPENDIO_OPERATORE check(STIPENDIO>0) ENABLE;
```

```
alter table OPERAIO  
add constraint STIPENDIO_OPERAIO check(STIPENDIO>0) ENABLE;
```

```
alter table MATERIALI  
add constraint COSTO_MASTERIALI check(COSTO>0) ENABLE;
```

```
ALTER TABLE AREA_CANTIERE  
ADD CONSTRAINT SINGOLO_TIPO_AREA_PER_CANTIERE UNIQUE(cod_area,cantiere)
```

```
ALTER TABLE MATERIALI  
ADD CONSTRAINT quantità_positiva check(quantità>0)
```

```
ALTER TABLE sensori
```

```
ADD CONSTRAINT sensori_diversi unique(tipo_sensore,area_cantiere)
```

## ***FUNZIONI***

```
CREATE OR REPLACE FUNCTION LIVELLO_CAPO_OPERAIO(cantiere in varchar2, capo_operai in varchar2)  
RETURN INT
```

```
IS
```

```
livello int;
```

```
BEGIN
```

```
SELECT o.livello_ruolo into livello from operaio o, cantiere c WHERE c.cod_cantiere=o.cantiere and  
c.cod_cantiere=cantiere and capo_operai=o.cf;
```

```
RETURN livello;
```

```
EXCEPTION
```

```
WHEN no_data_found THEN
```

```
RETURN 0;
```

```
END;
```

```
create or replace NONEDITIONABLE FUNCTION NUMERO_AREE(cantiere in varchar2) RETURN INT
```

```
IS numero_ree int;
```

```
begin
```

```
SELECT COUNT(cantiere) into numero_ree FROM AREA_CANTIERE_MASSIME a WHERE  
a.cantiere=cantiere GROUP BY CANTIERE;
```

```
RETURN numero_ree;
```

```
EXCEPTION
```

```
WHEN no_data_found THEN
```

```
RETURN 0;
```

```
END;
```

```
create or replace FUNCTION MACCHINE_DEL_CANTIERE(macchina in varchar2, cantiere_appartenenza in  
varchar2) RETURN VARCHAR2
```

```
IS appartiene VARCHAR2(16);
```

```
BEGIN
```

```
SELECT m.cod_macchina into appartiene from macchine m where m.cod_macchina=macchina and  
m.cantiere=cantiere_appartenenza;
```

```
RETURN appartiene;
```

```
EXCEPTION
```

```
when no_data_found then
```

```
RETURN NULL;
```

```
END;
```

## **TRIGGER**

```
CREATE OR REPLACE TRIGGER CANTIERE_APERTO_PER_OPERAIO
```

```
INSTEAD OF INSERT OR UPDATE ON OPERAIO_CANTIERE
```

```
FOR EACH ROW
```

```
DECLARE
```

```
data_apertura DATE;
```

```
data_chiusura DATE;
```

```
appartiene varchar2(16);
```

```
BEGIN
```

```
SELECT c.data_inizio, c.data_fine INTO data_apertura, data_chiusura
```

```
FROM cantiere c
```

```
WHERE c.cod_cantiere= :new.cantiere;
```

```
appartiene:= MACCHINE_DEL_CANTIERE(:NEW.MACCHINA,:NEW.CANTIERE);
```

```
IF(:NEW.data_assunzione>=data_apertura and :NEW.data_assunzione<data_chiusura ) THEN
```

```
dbms_output.put_line(appartiene);
```

```
IF(:NEW.livello_ruolo>=3 and :NEW.livello_ruolo<=5 and appartiene IS NOT NULL) THEN
```

```
DBMS_OUTPUT.put_line('AGGIUNTO CON SUCCESSO');
```

```
INSERT INTO operaio
VALUES(UPPER(:NEW.nome) , UPPER(:NEW.cognome), UPPER(:NEW.cf), :NEW.stipendio,
:NEW.data_assunzione, UPPER(:NEW.cantiere), UPPER(:NEW.macchina),:NEW.livello_ruolo);
ELSE
DBMS_OUTPUT.put_line('SOLO GLI OPERAI DAL LIVELLO 3 AL 5 POSSONO USARE LE MACCHINE
O MACCHINA DI UN ALTRO CANTIERE');
INSERT INTO operaio
VALUES(UPPER(:NEW.nome) , UPPER(:NEW.cognome), UPPER(:NEW.cf), :NEW.stipendio,
:NEW.data_assunzione, UPPER(:NEW.cantiere),NULL,:NEW.livello_ruolo);
END IF;
END IF;
END;
```

```
CREATE OR REPLACE TRIGGER CANTIERE_APERTO_PER_CAPOCANTIERE
INSTEAD OF INSERT OR UPDATE ON CAPOCANTIERE_CANTIERE
FOR EACH ROW
DECLARE
data_apertura DATE;
data_chiusura DATE;
BEGIN
```

```
SELECT c.data_inizio, c.data_fine INTO data_apertura, data_chiusura
FROM cantiere c
WHERE c.cod_cantiere= :new.cantiere;
IF(:NEW.data_assunzione>=data_apertura and :NEW.data_assunzione<data_chiusura ) THEN
INSERT INTO capocantiere
VALUES(UPPER(:NEW.nome) , UPPER(:NEW.cognome), UPPER(:NEW.cf), :NEW.stipendio,
:NEW.data_assunzione, UPPER(:NEW.cantiere), UPPER(:NEW.utente),UPPER(:NEW.password));
```

END IF;

END;

CREATE OR REPLACE TRIGGER max\_aree

INSTEAD OF INSERT OR UPDATE ON AREA\_CANTIERE\_MASSIME

FOR EACH ROW

DECLARE

numero int;

livello int;

BEGIN

    livello:=LIVELLO\_CAPO\_OPERAIO(:NEW.cantiere,:NEW.capo\_operai);

    numero:=NUMERO\_AREE(:New.cantiere);

    IF(livello>=6) THEN

        IF(3>numero) THEN

            INSERT INTO area\_cantiere

values(UPPER(:NEW.CODICE),UPPER(:NEW.capo\_operai),(:NEW.cantiere),UPPER(:NEW.tipo));

        ELSE

            DBMS\_OUTPUT.put\_line('MASSIMO NUMERO AREE RAGGIUNTO');

        END IF;

    ELSE

        DBMS\_OUTPUT.put\_line('SOLO UN OPERAIO DAL LIVELLO 6 PUÓ DIRIGERE UN AREA');

    END IF;

END;

CREATE OR REPLACE TRIGGER CANTIERE\_APERTO\_PER\_MATERIALI

INSTEAD OF INSERT OR UPDATE ON MATERIALI\_CANTIERE

FOR EACH ROW

DECLARE

data\_apertura DATE;

data\_chiusura DATE;

BEGIN

SELECT c.data\_inizio, c.data\_fine INTO data\_apertura, data\_chiusura

FROM cantiere c

WHERE c.cod\_cantiere= :new.cantiere;

IF(:NEW.data\_acquisto>=data\_apertura and :NEW.data\_acquisto<data\_chiusura ) THEN

INSERT INTO materiali

VALUES(UPPER(:NEW.tipo) , :NEW.costo, :NEW.data\_acquisto, UPPER(:NEW.codice),  
UPPER(:NEW.cantiere),UPPER(:NEW.fornitore),:NEW.quantità);

END IF;

END;

## **VISTE**

CREATE OR REPLACE VIEW OPERAIO\_CANTIERE("NOME", "COGNOME", "CF", "STIPENDIO",  
"DATA\_ASSUNZIONE", "CANTIERE", "MACCHINA", "LIVELLO\_RUOLO") AS

SELECT o.nome, o.cognome, o.cf, o.stipendio, o.data\_assunzione, o.cantiere, o.macchina, o.livello\_ruolo

FROM operaio o;

CREATE OR REPLACE VIEW CAPOCANTIERE\_CANTIERE("NOME", "COGNOME", "CF", "STIPENDIO",  
"DATA\_ASSUNZIONE", "CANTIERE", "UTENTE", "PASSWORD") AS

SELECT c.nome, c.cognome, c.cf, c.stipendio, c.data\_assunzione, c.cantiere, c.utente, c.password

FROM CAPOCANTIERE c;

```
CREATE OR REPLACE VIEW AREA_CANTIERE_MASSIME ("TIPO", "CODICE", "CAPO_OPERAIO", "CANTIERE")  
AS
```

```
SELECT "TIPO_AREA", "COD_AREA", "CAPO_OPERAIO", "CANTIERE" FROM area_cantiere;
```

```
CREATE OR REPLACE VIEW MATERIALI_CANTIERE ("TIPO", "COSTO", "DATA_ACQUISTO", "CODICE",  
"CANTIERE", "FORNITORE", "QUANTITÀ") AS
```

```
SELECT  
"TIPO_MATERIALE", "COSTO", "DATA_ACQUISTO", "COD_MATERIALE", "CANTIERE", "FORNITORE", "QUANTITÀ"  
FROM MATERIALI;
```