Si voglia progettare una base di dati per una piccola attività; nello specifico, in questo caso, trattasi di una ferramenta.

La base di dati deve fare in modo di semplificare la vita quotidiana del proprietario e dei suoi dipendenti, sostituendo il vecchio sistema informativo e costruendo un solido sistema informatico a supporto del tutto.

La base di dati deve ovviamente rispettare le proprietà ACID (Atomicità, Consistenza, Isolamento, Persistenza) e i dati stessi devono essere prontamente accessibili qualora ve ne sia il bisogno.

Le informazioni conservate devono essere quelle usuali di un esercizio quale la ferramenta, ovvero, il database deve memorizzare quelle relative ai:

* Dipendenti, che devono avere un determinato codice di riconoscimento, una mansione a cui sono assegnati, il progetto a cui partecipano (se affidato), un nome, un cognome ed una data di nascita.
* Materiali, che si dividono in materiali posseduti e materiali richiesti/in arrivo. Tutti i materiali devono possedere l’informazione sul tipo di materiale, la quantità ed il fornitore che attualmente lo procura. Sui materiali posseduti è bene sapere in che progetto sono attualmente utilizzati (se lo sono); dei materiali in arrivo, invece, si devono conservare i dati sulla data di arrivo e sulla data di spedizione.
* Macchinari, sui quali si devono memorizzare le informazioni sul codice della macchina, sul tipo della macchina, del dipendente che abitualmente lo usa e del progetto in cui è impegnato (se lo è).
* Progetti, i quali devono poter essere facilmente identificati da un codice identificativo, dei quali si conserva l’informazione sulla data di inizio, sulla data (prevista) di conclusione, sul nome (tipo) del progetto, sui materiali utilizzati, sulle macchine utilizzate (se il progetto necessita l’utilizzo di specifiche apparecchiature) e sui dipendenti che sono impegnati nello stesso.
* Clienti, dei quali si devono memorizzare i dati sul nome, cognome e il progetto che ha richiesto (se lo ha fatto).
* Fornitori, che devono possedere un nome ed il materiale che attualmente forniscono.
* Vendite, delle quali si vuole salvare l’oggetto venduto, a quale cliente, a che prezzo ed in quale data.

DIPENDENTI(Codice, Mansione, Nome, Cognome, DataNascita)

MACCHINARI(Codice, Tipo)

FORNITORI(Nome, MaterialeFornito)

MATERIALI\_RICHIESTI(Tipo, Quantità, DataInvio, DataArrivo, NomeFornitore:FORNITORI)

MATERIALI\_POSSEDUTI(Tipo, Quantità, NomeFornitore:FORNITORI)

PROGETTI(Codice, Tipo, Inizio, Fine)

CLIENTI(Nome, Cognome, CodiceProgetto:PROGETTI)

VENDITE(NomeCliente:CLIENTI, CognomeCliente:CLIENTI, Oggetto, DataVendita, Prezzo)

MANUALITA(CodiceDipendente:DIPENDENTI, CodiceMacchinario:MACCHINARI)

ASSEGNAZIONI(CodiceDipendente:DIPENDENTI, CodiceProgetto:PROGETTI)

ATTRIBUZIONI(CodiceMacchinario:MACCHINARI, CodiceProgetto:PROGETTI)

DOTAZIONI(TipoMateriale:MATERIALI\_POSSEDUTI, CodiceProgetto:PROGETTI)

---- RUOLI ----

Al fine di garantire un accesso che sia protetto, è bene definire alcuni ruoli all'interno della base di dati.

Incominciamo, ovviamente, col dichiarare il database administrator, il quale ha pieno controllo su ogni aspetto del database:

CREATE USER Ferramenta\_DBA IDENTIFIED BY admin;

GRANT dba TO Ferramenta\_DBA;

Una volta fatto ciò possiamo proseguire verso i ruoli più “limitati”.

Andando in ordine decrescente di permessi, definiamo il ruolo “amministrativo”. Esso servirà per garantire alcune operazioni su specifiche tabelle ai dipendenti che ricoprono cariche amministrative all'interno della nostra attività. Essi devono poter visualizzare, inserire, modificare, cancellare tuple all'interno delle tabelle relative ai macchinari, ai materiali richiesti, ai materiali posseduti, ai fornitori, ai clienti. Devono inoltre poter gestire la forza lavoro, per cui è necessario che essi possano assegnare dipendenti ad un determinato progetto e/o ad uno specifica apparecchiatura, oltre che a congedarli e/o sostituirli (oltre che, come di consueto, visualizzare quali dipendenti sono assegnati ad un macchinario o quali dipendenti stanno o sono in procinto di lavorare ad un progetto particolare).

A tale scopo quindi, esplicitiamo il ruolo:

CREATE ROLE Amministrativo;

GRANT CONNECT To Amministrativo;

GRANT SELECT, INSERT, UPDATE, DELETE ON Ferramenta\_DBA.MACCHINARI TO Amministrativo;

GRANT SELECT, INSERT, UPDATE, DELETE ON Ferramenta\_DBA.MATERIALI\_RICHIESTI TO Amministrativo;

GRANT SELECT, INSERT, UPDATE, DELETE ON Ferramenta\_DBA.MATERIALI\_POSSEDUTI TO Amministrativo;

GRANT SELECT, INSERT, UPDATE, DELETE ON Ferramenta\_DBA.FORNITORI TO Amministrativo;

GRANT SELECT, INSERT, UPDATE, DELETE ON Ferramenta\_DBA.CLIENTI TO Amministrativo;

GRANT SELECT, INSERT, UPDATE, DELETE ON Ferramenta\_DBA.MANUALITA TO Amministrativo;

GRANT SELECT, INSERT, UPDATE, DELETE ON Ferramenta\_DBA.ASSEGNAZIONI TO Amministrativo;

Passiamo adesso al secondo ruolo, quello del classico dipendente. Esso deve poter visualizzare, aggiungere, modificare o cancellare dei progetti. I dipendenti devono inoltre poter stabilire quali materiali e quali macchine sono più adeguate ad uno specifico progetto.

Dichiariamo quindi il ruolo Dipendente:

CREATE ROLE Dipendente;

GRANT CONNECT TO Dipendente;

GRANT SELECT, INSERT, UPDATE, DELETE ON Ferramenta\_DBA.PROGETTI TO Dipendente;

GRANT SELECT, INSERT, UPDATE, DELETE ON Ferramenta\_DBA.ATTRIBUZIONI TO Dipendente;

GRANT SELECT, INSERT, UPDATE, DELETE ON Ferramenta\_DBA.DOTAZIONI TO Dipendente;

Infine, occorre definire il ruolo del cassiere. Il cassiere deve unicamente lavorare sulla tabella vendite, in modo da poter gestire gli acquisti effettuati dai clienti dell'attività. Quindi:

CREATE ROLE Cassiere;

GRANT CONNECT TO Cassiere;

GRANT SELECT, INSERT, UPDATE, DELETE ON Ferramenta\_DBA.VENDITE TO Cassiere;

---- CREATE TABLE ----

CREATE TABLE DIPENDENTI(

Codice CHAR(8),

Mansione VARCHAR2(20),

Nome VARCHAR2(25),

Cognome VARCHAR2(25),

DataNascita DATE,

PRIMARY KEY(Codice)

);

CREATE TABLE MACCHINARI(

Codice CHAR(5),

Tipo VARCHAR2(25),

PRIMARY KEY(Codice)

);

CREATE TABLE FORNITORI(

Nome VARCHAR2(25),

MaterialeFornito VARCHAR2(25),

PRIMARY KEY(Nome)

);

CREATE TABLE MATERIALI\_RICHIESTI(

Tipo VARCHAR2(25),

Quantità INTEGER,

DataInvio DATE,

DataArrivo DATE,

NomeFornitore VARCHAR2(25),

PRIMARY KEY(Tipo),

FOREIGN KEY(NomeFornitore) REFERENCES FORNITORI(Nome)

ON DELETE SET NULL

);

CREATE TABLE MATERIALI\_POSSEDUTI(

Tipo VARCHAR2(25),

Quantità INTEGER,

NomeFornitore VARCHAR2(25),

PRIMARY KEY(Tipo),

FOREIGN KEY(NomeFornitore) REFERENCES FORNITORI(Nome)

ON DELETE SET NULL

);

CREATE TABLE PROGETTI(

Codice CHAR(4),

Tipo VARCHAR2(25),

Inizio DATE,

Fine DATE,

PRIMARY KEY(Codice)

);

CREATE TABLE CLIENTI(

Nome VARCHAR2(25),

Cognome VARCHAR2(25)

CodiceProgetto CHAR(4),

PRIMARY KEY(Nome, Cognome),

FOREIGN KEY(CodiceProgetto) REFERENCES PROGETTI(Codice)

ON DELETE SET NULL

);

CREATE TABLE VENDITE(

NomeCliente VARCHAR2(25),

CognomeCliente VARCHAR2(25),

Oggetto VARCHAR2(30),

DataVendita DATE,

Prezzo INTEGER,

PRIMARY KEY(NomeCliente, CognomeCliente),

FOREIGN KEY(NomeCliente) REFERENCES CLIENTI(Nome) ON DELETE SET NULL,

FOREIGN KEY(CognomeCliente) REFERENCES CLIENTI(Cognome)

ON DELETE SET NULL

);

CREATE TABLE MANUALITA(

CodiceDipendente CHAR(8),

CodiceMacchinario CHAR(5),

PRIMARY KEY(CodiceDipendente, CodiceMacchinario),

FOREIGN KEY(CodiceDipendente) REFERENCES DIPENDENTI(Codice) ON DELETE SET NULL,

FOREIGN KEY(CodiceMacchinario) REFERENCES MACCHINARI(Codice)

ON DELETE SET NULL

);

CREATE TABLE ASSEGNAZIONI(

CodiceDipendente CHAR(8),

CodiceProgetto CHAR(4),

PRIMARY KEY(CodiceDipendente, CodiceProgetto),

FOREIGN KEY(CodiceDipendente) REFERENCES DIPENDENTI(Codice) ON DELETE SET NULL,

FOREIGN KEY(CodiceProgetto) REFERENCES PROGETTI(Codice)

ON DELETE SET NULL

);

CREATE TABLE ATTRIBUZIONI(

CodiceMacchinario CHAR(5),

CodiceProgetto CHAR(4),

PRIMARY KEY(CodiceDipendente, CodiceProgetto),

FOREIGN KEY(CodiceMacchinario) REFERENCES MACCHINARI(Codice) ON DELETE SET NULL,

FOREIGN KEY(CodiceProgetto) REFERENCES PROGETTI(Codice)

ON DELETE SET NULL

);

CREATE TABLE DOTAZIONI(

TipoMateriale VARCHAR2(25),

CodiceProgetto CHAR(4),

PRIMARY KEY(TipoMateriale, CodiceProgetto),

FOREIGN KEY(TipoMateriale) REFERENCES MATERIALI\_POSSEDUTI(Tipo) ON DELETE SET NULL,

FOREIGN KEY(CodiceProgetto) REFERENCES PROGETTI(Codice)

ON DELETE SET NULL

);

--- QUERY ---

Query n.1

Trovare il nome e il cognome di tutti i dipendenti della ferramenta:

SELECT Nome, Cognome

FROM DIPENDENTI

Query n.2

Trovare il tipo del macchinario il cui codice è “A4563”:

SELECT Tipo

FROM MACCHINARI

WHERE Codice=”A4563”

Query n.3

Trovare il tipo di tutti i progetti il cui codice inizia con la lettera D:

SELECT DISTINCT Tipo

FROM Progetti

WHERE Codice LIKE ‘D%’

Query n.4

Trovare i clienti dei quali la seconda lettera del nome è una ‘i’ e di cui il cognome termina per ‘e’. Ordinare poi il risultato in ordine decrescente per cognome (se si trovano due cognomi uguali, allora ordinare per nome):

SELECT DISTINCT \*

FROM CLIENTI

WHERE Nome=’\_i%’ AND Cognome=’%e’

ORDER BY Cognome, Nome DESC

Query n.5

Trovare il codice dei progetti a cui sta attualmente partecipando il dipendente con codice “FD520101”:

SELECT ASSEGNAZIONI.CodiceProgetto

FROM DIPENDENTI JOIN ASSEGNAZIONI ON DIPENDENTI.Codice=ASSEGNAZIONI.CodiceDipendente

WHERE DIPENDENTI.Codice=’FD520101’

Query n.6

Somma delle quantità di tutti i materiali in magazzino:

SELECT SUM(Quantità) AS SommaQuantità

FROM MATERIALI\_POSSEDUTI

Query n.7

Trovare i macchinari che sono impegnati nel progetto “AS83” ma che non sono utilizzati nel progetto “BC40”:

SELECT M.Codice, M.Tipo

FROM MACCHINARI M JOIN ATTRIBUZIONI A ON M.Codice=A.CodiceMacchinario

WHERE A.CodiceProgetto=’AS83’

EXCEPT

SELECT M.Codice, M.Tipo

FROM MACCHINARI M JOIN ATTRIBUZIONI A ON M.Codice=A.CodiceMacchinario

WHERE A.CodiceProgetto=’BC40’

Query n.8

Trovare qual è il prezzo medio di vendita dell’oggeto “Pila”:

SELECT AVG(Prezzo) AS MediaPrezzo

FROM VENDITE

WHERE Oggetti=’Pila’

Query n.9

Materiali in deposito di cui si possiede una quantità superiore alle 1500 unità (sapendo che si possiedono proprio 1500 unità del materiale Ottone):

SELECT Tipo

FROM MATERIALI\_POSSEDUTI

WHERE Quantità > ANY ( SELECT Quantità

FROM MATERIALI\_POSSEDUTI

WHERE Tipo=’Ottone’ )

Query n.10

Per ogni progetto, trovare quanti dipendenti ne partecipano:

SELECT P.Codice, P.Tipo AS Progetto, COUNT(A.CodiceDipendente) AS Dipendenti

FROM PROGETTO P JOIN ASSEGNAZIONI A ON P.Codice=A.CodiceDipendente

GROUP BY P.Tipo

--- TRIGGER –

Il prezzo di un oggetto non può essere negativo.

1: create or replace trigger prezzononminore

2: before insert on VENDITE

3: for each row

4: declare

5: errore exception;

6: begin

7: if :new.Prezzo < 0

8: then raise errore;

9: end if;

10: exception

11: when errore then raise\_application\_error(-20002,’Non puoi inserire un prezzo che sia negativo.’)

12: end;