

## Laboratorio di Basi di Dati

Gruppo numero \_\_\_\_\_

Membro 1 (Nome Cognome) \_\_\_\_\_

Membro 2 (Nome Cognome) \_\_\_\_\_

Membro 3 (Nome Cognome) \_\_\_\_\_

MAGAZZINO (COD\_PROD, QTA\_DISP, QTA\_RIORD)

RIORDINO (COD\_PROD, DATA, QTA\_ORD)

FK: COD\_PROD REFERENCES MAGAZZINO

## Esercitazione 6

Il Database Ordini è un database per la gestione di magazzino e riordini (scaricare schema e dati dal sito dell'insegnamento).

## PARTE 1

- ☐ 1.a) Creare una funzione `prelievo()` per la gestione dei prelievi da magazzino. Seguire le seguenti specifiche e utilizzare il template sotto riportato riguardante la sintassi di PostgreSQL:
- L'utente indica un prelievo dando il codice del prodotto e la quantità da prelevare
  - Viene eseguito il prelievo, modificando la quantità disponibile in magazzino
  - La funzione restituisce la nuova quantità disponibile

Codice PL/pgSQL

```
CREATE FUNCTION prelievo (PROD INTEGER, QUANT INTEGER) RETURNS INTEGER AS $$
DECLARE
```

BEGIN

```
RETURN          ;
END;
$$ LANGUAGE 'plpgsql';
```

- ☐ 1.b) Creare un trigger gestione\_riordino (**evento**: modifica della tabella magazzino, **condizione**: nuova quantità disponibile inferiore a quantità di riordino) per gestire automaticamente i seguenti aspetti:
- Se si tenta di modificare la quantità disponibile con un numero negativo (la quantità disponibile in magazzino non è sufficiente per un prelievo) il trigger si arresta con una eccezione
  - Altrimenti, si predispone un nuovo ordine d'acquisto.

Codice PL/pgSQL

```
CREATE FUNCTION riordino() RETURNS trigger AS $$
    BEGIN

        RETURN NEW;
    END;
$$ LANGUAGE 'plpgsql';

CREATE TRIGGER GESTIONE_RIORDINO
AFTER          ON MAGAZZINO
FOR EACH ROW
WHEN (
EXECUTE PROCEDURE riordino();
```

- ☐ 1.c) Provare il funzionamento di funzione e relativo trigger, invocando la funzione di prelievo. Ad esempio:

```
SELECT prelievo(1,70);
SELECT prelievo(3,200);
```

- ☐ 1.d) Creare una funzione arrivo\_ordine() per gestire l'arrivo della merce riordinata a magazzino:

- L'utente indica il codice del prodotto arrivato
- In caso di prodotto non in riordino la procedura lancia un'eccezione
- Viene modificata la quantità disponibile in magazzino del prodotto (rispetto alla quantità di riordino di quel prodotto)
- Viene cancellata la tupla dalla tabella riordino
- La funzione restituisce la nuova quantità disponibile

Codice PL/pgSQL

```
CREATE FUNCTION arrivo_ordine(PROD INTEGER) RETURNS INTEGER AS $$
    DECLARE

    BEGIN

        END;
$$ LANGUAGE 'plpgsql';
```

- ☐ 1.e) Provare il funzionamento della funzione, ad esempio:

```
SELECT arrivo_ordine(1);
SELECT arrivo_ordine(4);
SELECT arrivo_ordine(1);
```