

# Basi di dati e Sistemi Informativi A.A. 2018-2019

## VI Esercitazione in laboratorio

### Selezioni su Premiere che utilizzano funzioni per il controllo del flusso

1. Elencare la descrizione di tutti gli articoli con indicazione esplicita della categoria di appartenenza (ad esempio `casalinghi` al posto di `cs`). Ordinare rispetto a categoria e descrizione.
2. Scrivere una interrogazione che restituisca, per ogni ordine, il numero totale di articoli in ogni categoria, nel seguente formato:

NumOrdine	Elettrodomestici	Casalinghi	ArticoliSportivi
12489	0	11	0
12491	2	0	0
12494	0	0	4
12495	0	2	0
12498	0	0	6
12500	1	0	0
12504	0	0	2

### Primi passi con procedure e funzioni su Premiere

1. La seguente procedura prende in ingresso una categoria di articoli e restituisce la descrizione degli articoli corrispondenti.

```
DELIMITER $$
DROP PROCEDURE IF EXISTS DescrizioneArticoli $$
CREATE PROCEDURE DescrizioneArticoli(c CHAR(2))
SELECT descrizione
FROM articoli
WHERE categoria = c $$
DELIMITER ;
```

```
CALL DescrizioneArticoli('el');
```

2. La seguente procedura prende in ingresso il nome e il cognome di un cliente e seleziona la descrizione e la quantità degli articoli eventualmente ordinati. La procedura inoltre restituisce in un parametro di uscita il numero totale di articoli ordinati dal cliente.

```
DELIMITER $$
DROP PROCEDURE IF EXISTS ArticoliCliente $$
```

```

CREATE PROCEDURE ArticoliCliente(in n char(8), in c char(10), out qta int)
begin
declare tmp int;
select nome,cognome,descrizione,qtaord
from Articoli natural join Dettagliordini natural join ordini
natural join Clienti
where nome=n and cognome=c;
select sum(A.qtaord) into tmp
from (select descrizione,qtaord
from Dettagliordini natural join Articoli natural join Clienti
natural join Ordini
where nome=n and cognome=c) as A;
set qta=tmp;
end $$
DELIMITER ;

-- START TRANSACTION;
CALL ArticoliCliente('Sally','Adams',@x);
select @x;

```

3. La seguente funzione restituisce la stringa 'My name is Premiere', nessun parametro in ingresso.

```

DELIMITER $$
DROP FUNCTION IF EXISTS Provaf $$
CREATE FUNCTION Provaf() RETURNS VARCHAR(30)
RETURN 'My name is Premiere' $$
DELIMITER ;

```

```

SELECT Provaf();

```

4. Periodicamente il prezzo degli articoli presenti nel db **Premiere** può subire un aumento. Scrivere una procedura che dati in ingresso 1) un parametro stringa e 2) l'aumento percentuale, aggiorni automaticamente i campi della tabella **Articoli** e visualizzi il contenuto della tabella così modificata secondo i seguenti criteri: la procedura deve aggiornare un articolo, se il primo parametro contiene la descrizione di un articolo, deve aggiornare tutti gli articoli di una categoria, se il primo parametro contiene la categoria da aggiornare e, infine, deve aggiornare tutti gli articoli, se il primo parametro è all.
5. Scrivere un procedura che dato in ingresso il parametro **data** elimini dalle tabelle **Ordini** e **DettagliOrdini** tutti i riferimenti ad ordini effettuati prima di quella data e memorizzi questi valori nelle tabelle **BackupOrdini** e **BackupDettagli**, dopo averle create, in modo che cancellazioni successive si aggiungano alle precedenti nelle tabelle di backup.

- \* Scrivere una procedura che permetta di inserire nuovi clienti nella tabella **Clienti**. Nella procedura deve essere definito un **handler** di tipo **exit** per l'errore **SQLSTATE '23000'** (Cannot add or update a child row: a foreign key constraint fails) in modo da gestire la violazione del vincolo di integrità referenziale tra la tabella **Clienti** e la tabella **Rappresentanti**.
- \* Lo stato di errore dell'esercizio 5) corrisponde anche al codice di errore MySQL numero 1452. Modificare la procedura del punto 5) facendo riferimento a questo codice invece che al codice **SQLSTATE**.
- \* In MySQL è possibile assegnare un qualsiasi altro nome ad uno stato di errore **SQLSTATE** o ad un codice di errore MySQL. La sintassi è:

```
DECLARE condition_name CONDITION FOR condition_value;
condition_value:  SQLSTATE sqlstate_value | mysql_error_code
```

Modificare la procedura del punto 5) definendo l'**handler** dopo aver assegnato allo stato di errore **SQLSTATE '23000'** il nome '**Violazione vincolo**'.

## Trigger su Premiere

1. Definire i trigger **Check\_prezzo\_ins** e **Check\_prezzo\_upd** sulla tabella **Articoli** che assegnino valore 0 al **PrzUnitario** qualora si tenti di inserire o aggiornare tale campo con un valore negativo.
2. Si vuole tenere un'anagrafica di tutte le scritture sulla tabella **Articoli**. Si definisca la tabella **LogArt** con i seguenti attributi:
  - un identificatore univoco;
  - data e ora della scrittura;
  - identificativo univoco del record cui si riferisce la scrittura;
  - operazione effettuata: I=Inserimento, M=Modifica, C=Cancellazione.

Definire quindi i trigger **Ana\_Art\_I**, **Ana\_Art\_U**, **Ana\_Art\_D** che inseriscano una nuova riga nella tabella **LogArt** in corrispondenza di operazioni di inserimento, aggiornamento e cancellazione sulla tabella **Articoli** (si usi la funzione **NOW()** per inserire data e ora della scrittura).