**SUB QUERY BASATE SU QUERY VETTORIALI E/O TABELLARI**

**‘Nome e il prezzo della rosa più costosa”.**

* **Tabelle necessarie, PRODUCTS CATEGORIES**

**SELECT ProductName, UnitPrice**

**FROM PRODUCTS INNER JOIN CATEGORIES ON CATEGORIES.ID = PRODUCTS.CategoryID**

**WHERE CategoryName = ‘Roses’ AND Unitprice = (**

**SELECT MAX(UnitPrice)**

**FROM PRODUCTS INNER JOIN CATEGORIES ON**

**CATEGORIES.ID = PRODUCTS.CategoryID**

**WHERE CategoryName = ‘Roses’**

**~~GROUP BY CategoryName~~ ‘INUTILE PERCHE’ HO GIA’ FILTRATO SULLE ROSE**

**)**

**‘Nome e il prezzo di tutti i prodotti a catalogo che costano di più della rosa più economica’**

**SELECT ProductName, UnitPrice**

**FROM PRODUCTS**

**WHERE Unitprice > (**

**SELECT MIN(UnitPrice)**

**FROM PRODUCTS INNER JOIN CATEGORIES ON**

**CATEGORIES.ID = PRODUCTS.CategoryID**

**WHERE CategoryName = ‘Roses’**

**)**

**SUB QUERY BASATE SU QUERY VETTORIALI SFRUTTANO I SEGUENTI OPERATORI**

* **ANY 🡪 La condizione in WHERE soddisfatta da almeno uno dei valori restituiti dalla query scalare**
* **ALL 🡪 La condizione deve essere soddisfatta da tutti**
* **IN 🡪 Il valore che cerco deve appartenere all’insieme dei valori restituiti dalla query vettoriale**

**‘Nome e il prezzo di tutti i prodotti a catalogo che costano di più della rosa più economica utilizzando l’operatore ANY’**

**SELECT ProductName, UnitPrice**

**FROM PRODUCTS**

**WHERE Unitprice > (**

**SELECT UnitPrice**

**FROM PRODUCTS INNER JOIN CATEGORIES ON**

**CATEGORIES.ID = PRODUCTS.CategoryID**

**WHERE CategoryName = ‘Roses’**

**)**

**OPERATORI DI ALGEBRA BOOLEANA INTERSEZIONE E DIFFERENZA VENGONO REALIZZATI NEL MODO SEGUENTE:**

* **DIFFERENZA (operatore di base)** 
  + **SQL EXCEPT**
  + **IN ACCESS (E IN SQL) EXCEPT SI OTTIENE CON <> ALL**

**SELECT \***

**FROM T1**

**EXCEPT**

**SELECT \***

**FROM T2**

**SELECT ID, Nome**

**FROM T1**

**WHERE Nome <> ALL (SELECT Nome FROM T2) AND**

**ID <> ALL (SELECT ID FROM T2)**

* **INTERSEZIONE**
  + **SQL 🡪 INTERSECT**
  + **ACCESS = ANY**

**OPERATORE EXIST 🡪 OPERATORE DI TIPO TABELLARE**

* **SCRIVO UNA QUERY ESTERNA**
* **LA QUERY VIENE ESEGUITA**
* **PER OGNI RECORD RESTITUITO DALLA QUERY ESTERNA VIENE ESEGUITA LA SUBQUERY**
* **LA SUBQUERY USA COME FILTRO UNO O PIU’ VALORI DEL RECORD RESTITUITO DALLA QUERY ESTERNA**

**QUERY ESTERNA**

|  |  |  |
| --- | --- | --- |
| **C1** | **C2 (filtro)** | **C3** |
| **~~…~~** | **~~2~~** | **~~…~~** |
| **…** | **3** | **…** |
| **…** | **4** | **…** |

**SELECT \***

**FROM TX**

**WHERE ID = C2**

**Se per esempio in TX ci fossero tutti gli ID a partire da 3**

**CERCO I CLIENTI CHE HANNO EFFETTUATO ALMENO UN ORDINE A PARTIRE DA UNA CERTA DATA**

**SELECT \***

**FROM CUSTOMERS**

**WHERE EXIST (**

**SELECT \***

**FROM CUSTOMERS INNER JOIN ORDERDETAILS ON CUSTOMERS.ID**

**= ORDERDETAILS.CustomerID**

**WHERE ORDERDETAILS.Date >= #01/01/2010#**

**AND ORDERDETAILS.CustomerID = CUSTOMER . ID**

**)**

|  |  |
| --- | --- |
| **ID CLiente** | **Cognome** |
| **1** | **AAA** |
| **2** |  |
| **3** |  |
| **4** |  |
| **5** |  |
| **6** |  |

**SELECT ID, DMax(“UnitPrice”, “PRODUCTS”)**

**FROM Products**

**GENERALMENTE EXIST SI USA NEGATO, NOT EXIST**

**Esempio “Categorie senza alcun prodotto ad esse assegnate”**

**SELECT CategoryName**

**FROM CATEGORIES**

**WHERE NOT EXISTS (SELECT \* FROM PRODUCTS WHERE PRODUCTS.CategoryID = CATEGORIES.ID)**

**Analogamente posso usare una Left Join**

**SELECT CategoryName**

**FROM CATEGORIES LEFT JOIN PRODUCTS ON CATEGORIES.ID = PRODUCTS.CageotryId**

**WHERE ISNULL(ProductID)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **CategoryName** | **…** | **ID** | **Productname** |
| **1** | **Nome Cate** |  | **1** |  |
| **1** |  |  | **2** |  |
| **1** |  |  | **3** |  |
|  |  |  |  |  |
| **100** | **Nuova Categoria** |  | **Null** |  |
|  |  |  |  |  |

**QUERY DI INSERIMENTO – CANCELLAZIONE – AGGIORNAMENTO**

**QUERY DI INSERIMENTO SONO DI 2 TIPI**

* **INSERT INTO**
* **SELECT INTO**

**INSERT INTO <Nome Tabella> [<(lista campi separati da virgola)>]**

**VALUES <(lista valori separati da virgola)>**

**INSERT INTO T1 (nome, età, data\_iscrizione)**

**VALUES (‘Abc’, 12, #01/01/2001#)**

**INSERT INTO T1 (nome, età, data\_isctrizione)**

**SELECT nome, età, data**

**FROM T2**

**WHERE ID > 10**

**SELECT <lista campi> INTO <Nome Tabella>**

**FROM <lista tabelle>**

**WHERE <lista condizioni>**

**AGGIORNAMENTO 🡪 UPDATE**

**UPDATE <Nome Tabella>**

**SET <nome campo> = <valore campo> [, <nome campo> = <valore campo>, …]**

**WHERE <lista condizioni>**

**UPDATE Customers**

**SET Nationality = ‘IT’**

**WHERE Nationality LIKE ‘[Ii][Tt]\*’**

**CANCELLAZIONE 🡪 DELETE**

**DELETE FROM <nome tabella>**

**WHERE <condizioni>**