**Lezione 7**

**LE INTERROGAZIONI SQL**

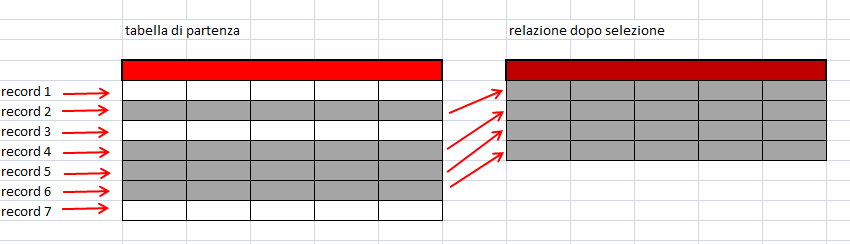
**(SELEZIONI)**

In questa lezione parleremo delle operazioni relazionali che si possono fare sulle tabelle. Ne esistono di tre tipi (**selezioni**, **proiezioni** e **congiunzioni**).

Le operazioni relazionali consentono di effettuare le interrogazioni alla base di dati per ottenere le informazioni desiderate estraendo da una tabella una sottotabella, oppure combinando tra loro due o più tabelle e generando così nuove relazioni.

Inizieremo dalle **selezioni** su una tabella.

La selezione genera una nuova relazione costituita da n-uple della relazione di partenza che soddisfano ad una determinata condizione; vengono, cioè, selezionate intere righe con i valori degli attributi corrispondenti alla condizione prefissata.



L’istruzione in SQL per operare una selezione su una tabella è la SELECT.

Nella sua forma generale l’istruzione è la seguente:

SELECT ALL \*

FROM *nome\_tabella*

WHERE *condizione;*

Dopo la clausola **WHERE** nella condizione possiamo scegliere tra vari tipi di operatori.

Gli operatori, messi a disposizione dal SQL standard si dividono in tre categorie:

• **Operatori di confronto**

**• Operatori aritmetici**

**• Operatori logici**

**Operatori di confronto**

Gli operatori di confronto servono a determinare uguaglianze e disuguaglianze tra valori e ad

effettuare ricerche all'interno dei dati. Di seguito uno schema tabellare:

|  |  |
| --- | --- |
| **=** | **Esprime uguaglianza tra due valori numerici** |
| **LIKE** | **Esprime somiglianza tra due valori letterali: con l'operatore LIKE è possibile usare,**  **per i confronti, i caratteri speciali \* (sostituisce un arbitrario numero di lettere)**  **e ? (sostituisce un arbitraria lettera)** |
| **<** | **Stabilisce se un valore è minore di un altro** |
| **>** | **Stabilisce se un valore è maggiore di un altro** |
| **<=** | **Stabilisce se un valore è minore o uguale di un altro** |
| **>=** | **Stabilisce se un valore è maggiore o uguale di un altro** |
| **<>** | **Stabilisce se due valori sono diversi tra loro** |
| **BETWEEN … AND …** | **Recupera un valore compreso tra due valori (es. BETWEEN 1°valore AND 2°valore)**  **Per valore si intende una costante o un campo del record.** |
| **IN** | **Stabilisce se un valore è contenuto in una lista di valori possibili** |
| **EXISTS** | **Stabilisce se un determinato record esiste** |

**Operatore LIKE**

L'operatore LIKE è alla base dei criteri di ricerca per parole chiave e delle query effettuate da un

motore di ricerca.

La sintassi di Microsoft Access è leggermente diversa dallo standard internazionale, e più

completa. Invece del simbolo underscore, si utilizza "?" per ricercare un singolo carattere (sia

numero che lettera); invece del simbolo "%", l'operatore jolly è dato da "\*".

**Operatori aritmetici**

Gli operatori aritmetici effettuano calcoli all'interno di una ricerca. Di seguito uno schema

tabellare :

Operatore Descrizione

|  |  |
| --- | --- |
| **+** | **Effettua un'addizione** |
| **-** | **Effettua una sottrazione** |
| **\*** | **Effettua una moltiplicazione** |
| **/** | **Effettua una divisione** |
| **\** | **Effettua una divisione intera (prende la parte intera senza i decimali)** |
| **^** | **Effettua un elevamento a potenza** |
| **sqr** | **Effettua una radice quadrata** |
|  |  |

**Operatori logici**

Gli operatori logici principali di SQL sono:

|  |  |
| --- | --- |
| **AND** | **e** |
| **OR** | **o oppure** |
| **NOT** | **non** |

Possono essere ripetuti più volte all'interno dell'elenco delle condizioni di ricerca.

L'operatore logico **AND**, che lega due (o più) condizioni, serve ad indicare alla ricerca di restituire

tutte le tuple (record) per le quali le due (o più) condizioni espresse sono tutte vere

contemporaneamente.

L'operatore logico **OR**, che lega due (o più) condizioni, serve ad indicare alla ricerca di restituire

tutte le tuple per le quali anche soltanto una delle due (o più) condizioni espresse è vera.

L'operatore **NOT** inverte il valore ad esso associato: se è vero restituisce un valore falso, mentre se

è falso restituisce un valore vero.

L'uso delle parentesi tonde ( ) all'interno dell'elenco delle condizioni di ricerca consente di

combinare condizioni differenti utilizzando entrambi i tipi di operatori logici per creare condizioni complesse.

Ipotizzando che la tabella si chiami **CLIENTE** e che volessimo estrarre solo i record degli agenti che operano nella provincia di Milano

**Cliente**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CodiceCliente** | **RagioneSociale** | **PartitaIva** | **Indirizzo** | **Provincia** | **CodiceAgente** |
| **A01** | **ROSSI** | **A08** | **Via x** | **Milano** | **B112** |
| **A02** | **BIANCHI** | **A09** | **Via y** | **Roma** | **B112** |
| **A03** | **NERI** | **A07** | **Via z** | **Milano** | **B112** |

L’istruzione sarà:

**SELECT ALL \***

**FROM CLIENTE**

**WHERE PROVINCIA = ”milano”;**

la relazione risultante sarà:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CodiceCliente** | **RagioneSociale** | **PartitaIva** | **Indirizzo** | **Provincia** | **CodiceAgente** |
| **A01** | **ROSSI** | **A08** | **Via x** | **Milano** | **B112** |
| **A03** | **NERI** | **A07** | **Via z** | **Milano** | **B112** |

Per scrivere l’istruzione in sql bisogna utilizzare l’editor di Access.

Esercizio:

Data la seguente tabella di nome FILM del database CINEMA



vogliamo ottenere:

1. Tutte le righe della tabella di genere “Drammatico”;
2. Tutti i film prodotti nell’anno 1960;
3. Tutti i film compresi tra gli anni 1948 e 1960 compresi;
4. Tutte le righe della tabella;
5. Tutte le righe dei film che hanno superato l’incasso di 10000 euro;
6. Solo le righe di genere “Teatro”;