

SQL – Oracle DB

- DBMS
 - Database relazionali: Oracle DB
 - Relazioni tra tabelle
- SQL
 - Amministrazione di database
 - Tipi di dato
 - Comandi (DML, DDL, ...)
 - SELECT
 - Operatori
- Progetto di riferimento
 - <https://github.com/egalli64/mjd> – folder oracle (*modulo 1*)

Database Management System

- Principali DBMS **Relazionali**
 - **Oracle**, MySQL, SQL Server, PostgreSQL, DB2
 - linguaggio standard (con variazioni): **SQL**
- NoSQL
 - MongoDB (doc), Redis (k-v), Neo4J (graph), Cassandra (wide-col)

<https://www.oracle.com/database/>

<https://www.oracle.com/database/technologies/xe-downloads.html>

<https://www.oracle.com/database/technologies/xe-prior-releases.html>

<https://www.oracle.com/database/technologies/appdev/jdbc.html>



Alcuni IDE per Oracle DB

- Quest Toad for Oracle
- Oracle SQL Developer
 - <https://www.oracle.com/tools/downloads/sqldev-downloads.html>
- Database Development per Eclipse
 - Help, Install New Software, Work with (...) → Database Development
 - Connessione per Oracle 11 XE → jdbc:oracle:thin:@localhost:1521:xe
- DBeaver (standalone o plugin per Eclipse)
- SQL * Plus
 - CLI (sqlplus.exe)

Database Relazionale

- **Tabella**

- Organizzata in **colonne**: un singolo **tipo** di dato (campo) memorizzato in una tabella
 - Ogni **riga** (o record) è una collezione di dati (colonne) di tipo che descrivono completamente un'entità
- Insieme di record persistente, o in memoria volatile (result set)
- Raccolte in uno **schema** del database, associato ad un **utente**
 - Un utente può avere il permesso di accedere tabelle di altri schemi

- Relazioni tra tabelle: **primary key** (PK) → **foreign key** (FK)

- PK: identifica univocamente (naturale o surrogata) una riga nella tabella corrente
- FK: identifica univocamente una riga in un'altra tabella

- Si possono stabilire delle regole (**constraint**) sulle colonne

- Il DBMS garantisce l'integrità di dominio (tipo di dato), PK, referenziale (FK → PK)

Relazioni tra tabelle

- **One to many / many to one**
 - Uno stato (PK) → molte città (FK duplicata)
- **Many to many** (implementato via tabella intermedia)
 - Uno stato → molte organizzazioni
 - Una organizzazione → molti stati
- **One to one**
 - Uno stato (PK) → una capitale (FK unique)

SQL

- DQL – Data Query Language
 - **SELECT**
- **DML** – Data Manipulation Language
 - INSERT, UPDATE, DELETE
- **DDL** – Data Definition Language
 - CREATE, ALTER, DROP, RENAME, TRUNCATE
- TC – Transaction Control
 - COMMIT, ROLLBACK, SAVEPOINT
- DCL – Data Control Language
 - GRANT, REVOKE

Le keyword SQL sono
case insensitive

select = SELECT

Amministrazione DBMS

- L'utente **root** ha tutti i privilegi per la gestione dell'istanza corrente di Oracle DB
 - sqlplus / as sysdba [il comando generico è sqlplus user/password@host]
 - Da Oracle 12 in avanti va specificato il container (orclpdb, xepdb1, ...) e poi aperto il database
 - alter session set container = xepdb1; alter database open;
- Attivazione di un utente (HR è lo storico utente/schema Oracle di esempio)
 - alter user hr identified by hr account unlock;
- I permessi di sysdba permettono di creare un utente e del suo schema associato
 - create user **me** identified by **password** account unlock;
 - grant connect, resource to **me**;
 - alter user **me** quota unlimited on users;
- Per eseguire uno script da SQL * Plus basta prefissarlo con '@'
 - Es: @migration

TNSNAME.ORA

- Nella subdirectory /network/admin della home del database

XE =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = xxx.xxx.xxx.xxx)(PORT = 1521))

(CONNECT_DATA =

(SERVER = DEDICATED)

(SERVICE_NAME = XE)

)

)

Indirizzo IP della macchina

Da Oracle 12:
nome del container
(orclpdb, xepdb1, ...)

- In caso di cambiamenti: refresh del listener Oracle

lsnrctl reload

- connessione via tnsname

sqlplus hr/hr@xe

Principali tipi di dato

DECIMAL(precision, scale)

INTEGER

CHAR(length)

VARCHAR2(length)

DATE

TIMESTAMP

SELECT

- Selezione di dati (colonne) da una tabella, filtrata per colonne e righe

```
select region_name from regions where region_id = 1;
```

- Selezione dei soli valori unici

```
select distinct manager_id from employees;
```

- Modifica i risultati in lettura da tabella

```
select job_title, min_salary, min_salary + 2000, min_salary * 3 + 1000 from jobs;
```

- Alias di colonna, introdotto da AS (opzionale) e delimitato da apici (singoli o doppi)

```
select job_title, min_salary + 2000 "increased min salary" from jobs;
```

- La tabella DUAL

```
select 1+2, 3-4, 2*6, 5/2, sysdate, current_date from dual;
```

- Concatenazione via operatore || (quello che per molti programmatori è l'OR)

```
select country_id || '...' || region_id || '!' from countries;
```

- Pseudo colonne

- select rowid, rownum from regions;

Dettagli su tabelle

- Tabelle dell'utente corrente

```
select table_name  
from user_tables;
```

- Descrizione di una tabella

```
select column_name, nullable, data_type, data_length, data_precision, data_scale  
from user_tab_columns  
where table_name = 'JOBS';
```

(in sqlplus e SQL Developer: describe jobs;)

NULL

- Valore non presente o non valido, check esplicito con “is null”

```
select first_name, last_name  
from employees  
where commission_pct is null;
```

- “Assorbe” altri operandi

```
select first_name, last_name, 12 * salary * commission_pct from employees;
```

- La funzione NVL() permette di decidere il comportamento

```
select first_name, last_name, 12 * salary * nvl(commission_pct, 0)  
from employees;
```

Operatori di confronto

=, !=, <, >, <=, >=, ANY (almeno uno), ALL

select * from regions where region_id = 1;

select * from regions where region_id != 2;

select * from regions where region_id < 3;

select * from regions where region_id <= 3;

select * from regions where region_id > any(1, 2, 3);

select * from regions where region_id > all(1, 2, 3);

Operatori di confronto

LIKE, BETWEEN, IN, IS NULL. Per negare il loro risultato: **NOT**

- **LIKE** wildcard: `_ %`

select last_name from employees where last_name like '_ul%';

- **BETWEEN**

select * from regions where region_id between 2 and 3;

select * from countries where country_name between 'A' and 'C';

- **IN**

select * from regions where region_id not in (2, 3);

select * from regions where region_id not in (2, 3, null); -- !! NOT IN(..., NULL) → FALSE !!

- **IS NULL**

select * from employees where manager_id is null;

Operatori logici

- **AND**

```
select * from employees  
where salary < 3000 and employee_id > 195;
```

- **OR** (disgiunzione inclusiva)

```
select * from employees  
where salary > 20000 or last_name = 'King';
```

- **NOT**

```
select * from employees  
where not department_id > 20;
```

Ordinamento via ORDER BY

- ORDER BY segue FROM – WHERE

```
select * from employees
```

```
order by last_name;
```

- ASC (ascending, default) / DESC (descending)

```
select * from employees
```

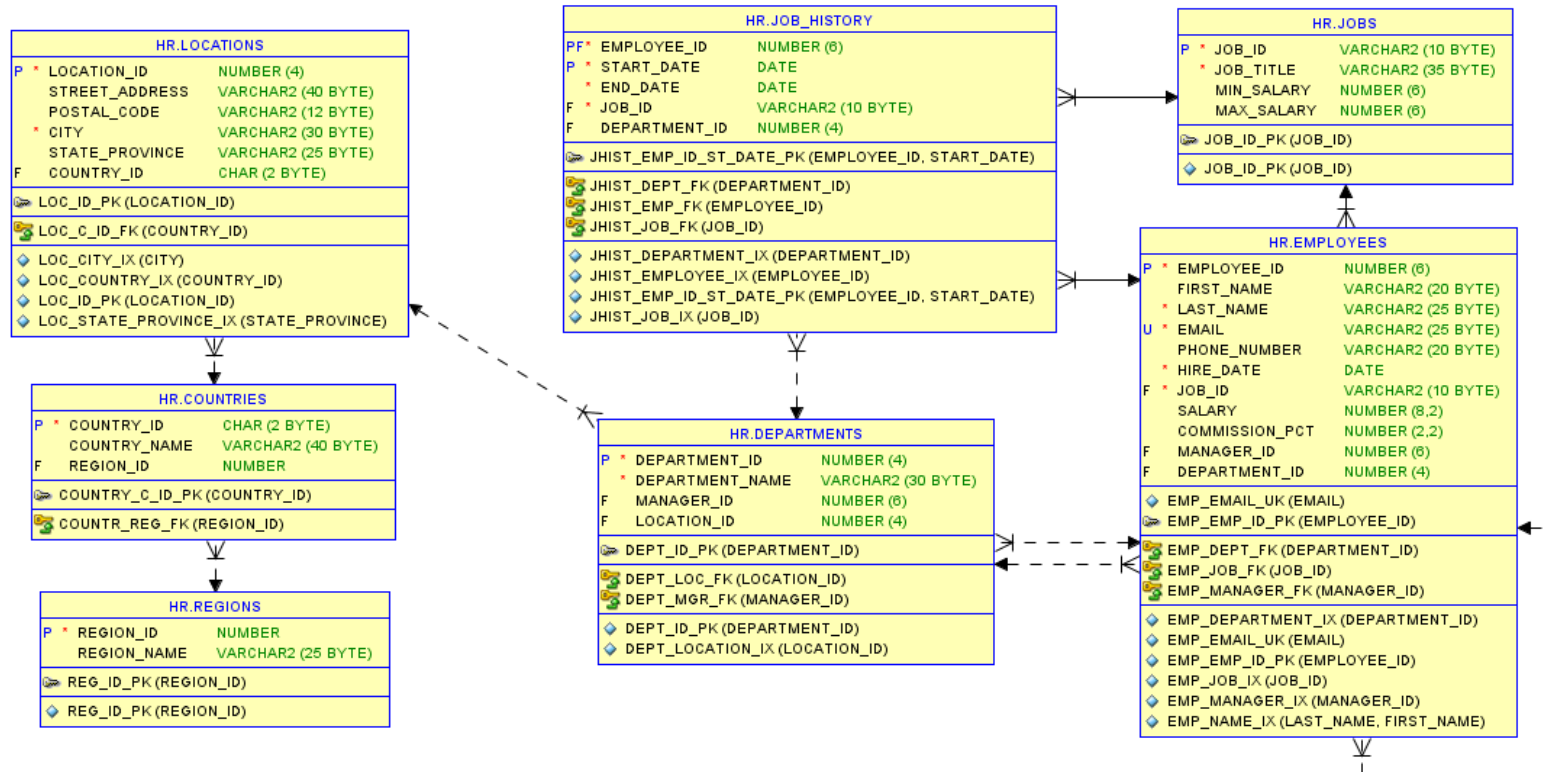
```
order by last_name desc, first_name asc;
```

- notazione posizionale

```
select first_name, last_name from employees
```

```
order by 2;
```


Diagramma Entity-Relation



Esercizi

- Employees: nome, cognome, email, telefono, data di assunzione
 - Tutti i dipendenti, ordinati per cognome e nome
 - Chi ha nome David o Peter
 - Chi appartiene al dipartimento 60
 - Chi appartiene ai dipartimenti 30, 50
 - Chi ha salario
 - maggiore di 10000
 - minore di 4000 o maggiore di 15000
 - minore di 4000 o maggiore di 15000, ma solo per i dipartimenti 50 e 80

Esercizi

- Employees
 - Chi è stato assunto nel 2005
 - Quali job_id sono presenti, in ordine naturale
 - Chi ha una commissione
 - Chi ha una 'a' nel nome o cognome
- Departments
 - Nomi, in ordine naturale
- Locations
 - Indirizzi delle sedi italiane