Database Relazionali

- Modello inventato da Edgar F. Codd nel 1970 premio Turing 1981
- Principali Database Management System (DBMS) Relazionali
 - Oracle, MySQL, SQL Server, PostgreSQL, DB2, SQLite, ...
- Linguaggio standard (con variazioni): SQL
- Definizione informale
 - I dati sono presentati agli utenti come tabelle
 - Sono forniti operatori relazionali per la loro gestione
- Non è l'unico modo per gestire basi di dati NoSQL
 - MongoDB (doc), Redis (k-v), Neo4J (graph), Cassandra (wide-col)

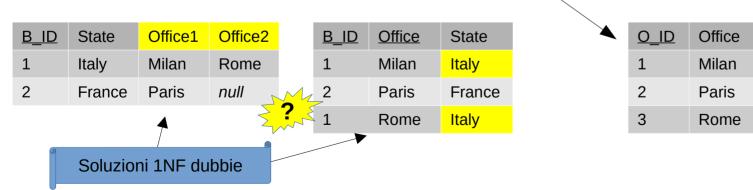
Modello Relazionale

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)
- Tipicamente le tabelle sono 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 (naturale o surrogata): identifica univocamente una riga nella tabella corrente
 - FK: identifica univocamente una riga in un'altra tabella
- Si possono stabilire delle regole (constraint) sulle colonne
- II DBMS garantisce l'integrità di dominio (tipo di dato), PK, referenziale (FK \rightarrow PK)

Normalizzazione – 1NF

- Strutturazione di un database allo scopo di ridurre la ridondanza, migliorare l'integrità dei dati
- 1NF prima forma normale
 - PK: ogni riga deve essere identificata univocamente
 - Non ci devono essere gruppi ripetuti
- Esempio, tabella B (branches), succursali di una azienda
 - La colonna Offices non è "atomica", le celle possono contenere più valori
 - Soluzione ottimale estraendo gli uffici in un'altra tabella



<u>P_ID</u>	State	Offices
1	Italy	Milan, Rome
2	France	Paris

B ID

B_ID State1 Italy2 France

2NF

- Seconda forma normale, è 1NF, e in più
 - ogni colonna che non è parte della PK deve dipendere dalla PK completa
 - Dunque, una tabella 1NF con PK <u>non composta</u> è anche 2NF
 - In caso di PK composta vanno verificate eventuali dipendenze
- Esempio, la tabella B, in una versione già vista come dubbia per 1NF
 - La colonna State dipende solo da B_ID, parte della PK
 - La soluzione ottimale proposta per 1NF è anche 2NF

B_ID	<u>Office</u>	State
1	Milan	Italy
2	Paris	France
1	Rome	Italy

O_ID	Office	B_ID
1	Milan	1
2	Paris	2
3	Rome	1

B_ID	State
1	Italy
2	France

3NF

- Terza forma normale, è 2NF e in più
 - Ogni colonna che non fa parte della PK dipende in modo non transitivo dalla PK completa
- Esempio: tabella di premi al miglior film europeo
 - PK composita, nome del premio e anno di assegnamento
 - È 2NF ma Director dipende transitivamente dalla chiave via Title
 - La soluzione sta nel creare due tabelle in relazione tra loro

Premi → Film

<u>Prize</u>	<u>Year</u>	Film_id
EFA	2013	1
Goya	2018	3
EFA	2015	2

<u>Prize</u>	<u>Year</u>	Title	Director
EFA	2013	La grande bellezza	Paolo Sorrentino
Goya	2018	The square	Ruben Östlund
EFA	2015	Youth	Paolo Sorrentino

Film_id	<u>Title</u>	Director
1	La grande bellezza	Paolo Sorrentino
2	Youth	Paolo Sorrentino
3	The square	Ruben Östlund

Relazioni tra tabelle

- Many to one / one to many
 - Molte città (FK duplicata) → uno stato (PK)
- One to one
 - Una città, ad es. la capitale politica (FK unique) → uno stato (PK)
- Many to many (implementato via tabella intermedia)
 - Uno stato → molte organizzazioni
 - Una organizzazione → molti stati

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