

Formulario

Op.	Cond.	Lett.	SQL
$\Pi_A(R)$		isolo colonne, senza duplicati	SELECT DISTINCT A FROM R
$\sigma_F(R)$		isolo righe che soddisfano F	SELECT * FROM R WHERE F
$R \times S$	schemi disgiunti	tutte le combinazioni di R e S	R CROSS JOIN S
$R \cup S$	stesso schema	tuple in R o S	R UNION S
$R \cap S$	stesso schema	tuple in R e in S	R INTERSECT S
$R - S$	stesso schema e grado	tuple in R ma non in S	EXCEPT o NOT IN/NOT EXISTS
$R \bowtie_F S$	schemi disgiunti	prodotto cartesiano con selezione	R JOIN S ON F
$R \bowtie S$	almeno un attributo in comune	prodotto cartesiano con selezione	R NATURAL JOIN S
$R \div S$	almeno un attributo in comune	tuple in R che compaiono in S	

Tips:

- La divisione $R \div S$ si può anche scrivere come $\Pi_D(R) - \Pi_D((\Pi_D(R) \times S) - R)$
- Parti sempre dalle sotto-query
- La divisione usala quando nella richiesta c'è una condizione del tipo "tutti"
- Se usi un operatore di aggregazione (COUNT, SUM, AVG, MAX, MIN), puoi usare GROUP BY e puoi usare HAVING

Normalizzazione tips:

- $X \rightarrow Y$: per una stessa X non ci sono Y diverse
- Se un attributo non compare mai a destra allora fa parte sicuramente della chiave
- E' BCNF se per ogni $A \rightarrow B$, A chiave e $B \not\subseteq A$
- E' 3NF se $A \rightarrow B$ con A chiave/superchiave **oppure** B attributo primo
- Se R non è in 3NF allora non è in BCNF
- Se R non è in BCNF potrebbe essere in 3NF

Decomposizione in 3NF:

- Individuare le chiavi candidate
 - Scrivere tutte le X^+ per ogni attributo a sinistra della freccia delle dipendenze funzionali fornite
 - Se ci sono attributi multipli, in X^+ vanno le dipendenze dei singoli attributi e le dipendenze degli attributi multipli
 - Ricorda di mettere anche la dipendenze derivate seguendo la regola sopra
 - Le chiavi candidate sono tutte quelle X^+ che contengono tutti gli attributi della tabella
- Scompongo in tabelle delle dipendenze usando le dipendenze funzionali
- Guardo se almeno una delle tabella ha come chiave la chiave della tabella originale
- Se la chiave si trova a sinistra è anche BCNF

SQL

- HAVING è utile per fare condizioni sugli attributi aggregati (*ad esempio se vogliamo avere tutti i gruppi con una media superiore a 8*)

- ALL: `SELECT * FROM R WHERE A > ALL (SELECT B FROM S)` restituisce le tuple di R che soddisfano la condizione $A > B$ se A è maggiore di tutti i valori di B
- ANY: simile a prima ma restituisce le tuple di R che soddisfano la condizione $A > B$ se A è maggiore di almeno un valore di B
- [NOT]EXISTS: serve per controllare se (non) esiste almeno una tupla in una sotto-query
- GROUP BY: raggruppa le tuple in base ai valori di un attributo
- VISTE: `CREATE VIEW <nome_vista> AS <sotto_query>`
- TABELLE TEMPORANEE: `WITH <nome_tabellatemp>(<nome_colonne>) AS (<sotto_query>)`