

PARCO GIOCHI → ESSEMPI

17/11 (SQL)

Prova di laboratorio: SQL Interrogazioni al database, creazione entità relazione. (Sub query - Query di selezione co

[ESSEMPI] ... → SCHEMA ER

VISITATORI → ROSSA MAGNETICA
(ID / COLORE)

ROSSA → OVER 14 F

BLU → OVER 14 M

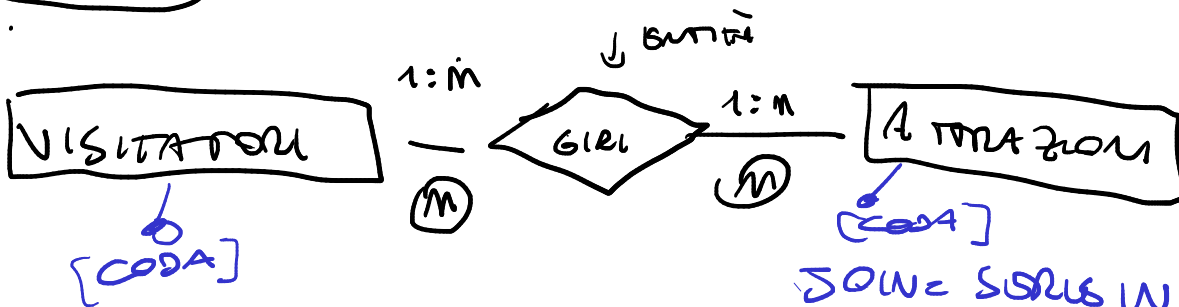
ROSSA → UNDER 14 F

ATTIRAZIONI → VISITATO(?)

NATURALIZZAZIONE

DISCRETE

SOLN



SOLN = SOLUS IN SQL

SUBSET → FROM VISITATORI AS VIS

SOLN ATTIRAZIONI AS AT ON AT.CODA = VIS.CODA

VISITATORI → ... / CODA
ATTIRAZIONI → ... / CODA
} COLLEGAMENTO (SOLN = RELAZIONE)

[VISITATORI]

- NOME SORA
- COGNOME

[GIRLI]

- DATA
- ORA
- NT (NUM. NASPI)
- NA (NUM. ATTRAZIONI)

[ATTRAZIONI]

- NOME
- DESCRIZIONE
- VISITATO

QUERY 1



INSERITA UNA DATA,
STABILIRE PER OGNI

ATTRAZIONE QUANTE VISITE

(= QUANTE VOLTE È STATO
VISITATO)

→ SELECT A.DESCRIZIONE, COUNT(NOME SORA) AS N-VISITE
→ FROM GIRLI G

WITH DATA = "2024-11-17"

JOIN VISITATORI V ON V.NOME SORA = G.NT
GROUP BY A.DESCRIZIONE

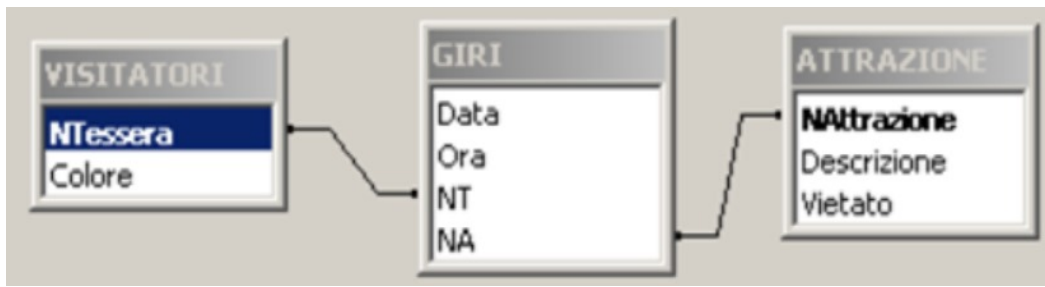
ACCESS

DESCR.	N. VISITE

→ COUNT = CONTIGLIO

QUANDO USIAMO COUNT, USIAMO ANCHE "GROUP BY"
PER INSERIRE LE COLONNE IN PIÙ

[VISITATORI
[NOME SORA]] — [GIRLI
[NT]] ⇒ SOLU!



QUERY 3 → STABILIRE [QUANTI] VISITATORI

HANNO VISITATO ATTRAZIONI < 14 ANNI

SELECT COUNT (NTESSERA) AS N-VISITING
FROM VISITATORI

JOIN GIRI ON VISITATORI.NT = GIRI.NT

JOIN ATTRAZIONI ON GIRI.NA < ATTRAZIONI.NATtrazione

WHERE VISITATO = TRUE.

QUERY 5 → ATTRAZIONI PIÙ VISITATE

[STAR = * = TUTTO] IN ASSOLUTO

SELECT COUNT (*) AS N-VISITING, A. DESCRIZIONE, A. NATtrazione
FROM ATTRAZIONI A

JOIN GIRI ON VISITATORI.NT = GIRI.NT

JOIN ATTRAZIONI ON GIRI.NA < ATTRAZIONI.NATtrazione

GROUP BY A. DESCRIZIONE, A. NATtrazione

QUERY 6 → QUANTI VISITATORI HANNO AVUTO ACCESSO
AL PARCO DI UNIVERSITÀ PER
TIPO DI TESSERA

```
[ SELECT V.Colore as TipoTessera, COUNT(*) as NumeroVisitatori
FROM VISITATORI V
GROUP BY V.Colore;
```

Colore	N. Visit
VERDE	10
BLU	15
ROSSO	20

AS = ALIAS -
RINOMINA TABELLE

→ TABOLA FINALE
(OUTPUT)

ASSICURAZIONI / SINISTRI

SCHEMA RELAZIONALE:

Proprietario (CodF, Nome, Cognome, Residenza)
 Assicurazione (CodAss, Nome, Sede)
 Sinistro (CodS, Localita, Data)
 Auto (Targa, Marca, Cilindrata, Potenza, CodF*, CodAss*)
 Autocoivolte (CodS*, Targa*, ImportoDelDanno)

CREARE TABELLE

CREAZIONE TABELLE + QUERY

Proprietario (CodF, Nome, Cognome, Residenza)
 CODF (INT) AUTO-
 NON NULL
 PRIMARY KEY,

DESCRIZIONE

Nome VARCHAR(50),

Cognome NOT NULL VARCHAR(50),

↑ NON NULLO = SI PUÒ METTERE

);

CREATE TABLE (TABELLA)
 Nome - campo tipo DESCRIZIONE

SINTASSI

SUBQUERY → QUERY ANNIDATA

ETA > DI ETA MEDIA



1. Subquery nel WHERE
 SELECT colonna
 FROM tabella
 WHERE colonna IN (SELECT colonna FROM altra_tabella);

2. Subquery nel FROM
 SELECT *
 FROM (SELECT * FROM tabella WHERE condizione) AS subquery;

3. Subquery nel SELECT
 SELECT colonna,
 (SELECT COUNT(*) FROM altra_tabella WHERE condizione) AS conteggio
 FROM tabella;

← QUESTA
 È
 ANCHE
 NON INVENTA
 USATA

-- Non Correlata (indipendente dalla query esterna)
 SELECT *
 FROM Clienti
 WHERE eta > (SELECT AVG(eta) FROM Clienti);

↓ STA CONSUMA ↓ STA RESIN

-- Correlata (dipende dalla query esterna)
 SELECT *
 FROM Ordini o
 WHERE importo > (SELECT AVG(importo) FROM Ordini WHERE cliente_id = o.cliente_id);

↓ IMPORTO
 UNICO

-- IN / NOT IN
 SELECT *
 FROM Prodotti
 WHERE categoria_id IN (SELECT id FROM Categorie WHERE tipo = 'Elettronica');

-- EXISTS / NOT EXISTS
 SELECT *
 FROM Clienti c
 WHERE EXISTS (SELECT 1 FROM Ordini WHERE cliente_id = c.id);

-- ANY / ALL
 SELECT *
 FROM Prodotti
 WHERE prezzo > ANY (SELECT prezzo FROM Prodotti WHERE categoria = 'Lusso');

∃ QUESTA VERA?

∃ AUTOMO IN ORDINI?

∃ PREZZO
 >
 TUTTI
 DI LUSO?

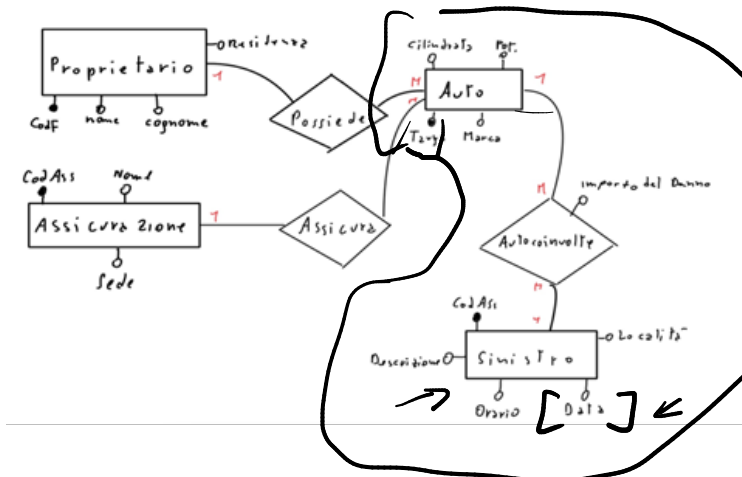
∃ =
 ESISTE

IN → ESISTE UNA
 CERTA COSA
 DENTRO UNA QUERY

EXISTS → ESISTE
 DENTRO UNA
 QUERY O
 LO STESSE

ANY/ALL →

NE ESISTE AUTOMATO
 ESISTE NO TUTTI?



NOT IN
 EXISTS

⑨ → TARGA AUTO NON CONNUOTA
 IN SINISTRA DOPO 20/01/2001

NOT IN (NON COINVOLTE IN SINISTRI)
 SELECT A.Targa FROM Autocoinvolte A
 WHERE DATA > '2001-01-20'
 JOIN SINISTRO S ON S.CodAss = A.CodAss

```
-- 9. Targa delle auto non coinvolte in sinistri dopo 20/01/2001 (2)
SELECT A.Targa
FROM Auto A
WHERE A.Targa NOT IN (
  SELECT AC.Targa
  FROM Autocoinvolte AC
  JOIN Sinistro S ON AC.CodS = S.CodS
  WHERE S.Data > '2001-01-20'
);
```

SOTTOQUERY
DOUS ESISTE
SOLO (2)

```
-- 7. Per auto coinvolte in più di un sinistro: targa, nome assicurazione e totale danni
SELECT A.Targa, AS.Nome, SUM(AC.ImportoDelDanno) as TotaleDanni
FROM Auto A
JOIN Autocoinvolte AC ON A.Targa = AC.Targa
JOIN Assicurazione AS ON A.CodAss = AS.CodAss
GROUP BY A.Targa, AS.Nome
HAVING COUNT(AC.CodS) > 1;
```

HAVING → IF SU
AGGREGAZIONI

CONDIZIONI → FAI UN IF
SU "COUNT"

ALTRE FUNZ. DI

AGGREGAZIONI ?

<https://alexchang7a.medium.com/sql-intermediate-cheatsheet-aggregation-c6c73b6831e9>

[CHEAT-SHEET SQL]

ORDINE DEL GLORIO: ARGOMENTI

QUERY CONDITIONS →

SELECT
FROM
WHERE

SOTTOQUERY →

IN / NOT IN / EXISTS /
NOT EXISTS

GROUP BY
HAVING
COUNT / SUM / MAX / MIN

