

Databases · supported focus

Supported focus · SL

Supported scope

- Supported SL focus. First assessment 2027 target; official PDF returns 403; older acquired brief is final assessment 2026. Remaining guide, assessment and practical requirements retain their recorded holds.
- State model conditions and distinguish fictional data from measurements.

Databases · 1.1

Reasoning

Duplicating a user address in every order can produce conflicting records when the address changes. Start with a prediction. State the quantities or features you would compare, then decide what evidence could distinguish two explanations.

Databases · 2.1

Reasoning

A relational table contains rows and attributes. A primary key identifies a row; a foreign key links to a referenced key in another table. Normalization can reduce avoidable duplication and update anomalies while preserving meaningful relationships.

primary key: Attribute set uniquely identifying a table row; foreign key: Attribute set referencing a key in another table.

Databases · 3.1

Reasoning

A join combines rows according to a specified condition. The result count depends on relationship cardinality and filters. A foreign key constraint enforces a relationship rule; it does not automatically encrypt personal data. Design a small synthetic user-order dataset with no real personal information. Declare keys, test duplicate and missing-reference inserts, then check a query result against a hand-worked expected table.

Databases · 4.1

Reasoning

State the known values and their units. Choose the relation because its assumptions fit this case, then rearrange before substitution. Known: four customers have respectively 2, 0, 3 and 1 orders. An inner join of customer to order returns $2+0+3+1 = 6$ rows. A left join also retains the customer with no order as one null-matched row, giving 7 here.

Databases · 4.2

Reasoning

Three customers have 1, 3 and 2 orders. How many rows does the customer-order inner join return? Use the same sequence: known quantities $\rightarrow$ model $\rightarrow$ relation $\rightarrow$ substitution $\rightarrow$ unit and interpretation.

Databases · 5.1

Reasoning

A primary key need not be a person name. A join is not a simple concatenation of tables. Use parameterized queries for untrusted input rather than building SQL from raw strings. Return to the original observation. Explain what the result supports, which conditions it assumes, and one way to test a competing explanation.

Databases · 5.2

Reasoning

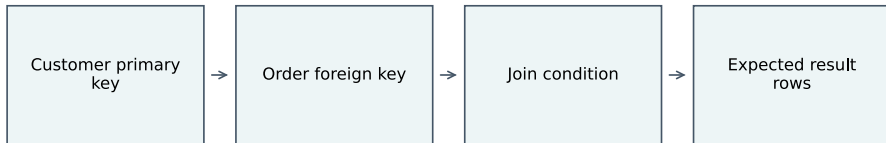
A foreign key automatically encrypts every row. This claim is false: A primary key need not be a person name. A join is not a simple concatenation of tables. Use parameterized queries for untrusted input rather than building SQL from raw strings.

Databases · 5.3

Reasoning

Relational databases: a key and a relationship: A join combines rows according to a specified condition. The result count depends on relationship cardinality and filters. A foreign key constraint enforces a relationship rule; it does not automatically encrypt personal data.

Databases · diagram



- Original teaching diagram: Relational databases: a key and a relationship
- Use the labels and the stated model together.

Databases · original transfer

Your turn

Customers A,B,C have 2,0,1 orders. Compare customer-order inner and left-join row counts. Why can counting joined rows miscount customers?

Databases · transfer reasoning

Method and interpretation

Inner join:3 rows; left join:4 including a null match for B. COUNT(*) counts rows; COUNT(order_id) excludes null; DISTINCT customer_id counts represented customers. State keys and join conditions.

Databases · executable trace

- Trace the state changes and boundary guard.
- Run the boundary case and compare the printed output.

```
import sqlite3
db = sqlite3.connect(":memory:")
db.executescript("""CREATE TABLE c(id INTEGER PRIMARY KEY);
CREATE TABLE o(id INTEGER PRIMARY KEY, cid INTEGER);
INSERT INTO c VALUES(1),(2),(3);
INSERT INTO o VALUES(10,1),(11,1),(12,3);""")
for kind in ["INNER", "LEFT"]:
    sql = f"SELECT COUNT(*),COUNT(o.id) FROM c {kind} JOIN o ON c.id=o.cid"
    print(kind, db.execute(sql).fetchone())
db.close()
```

Check yourself

- Explain primary key using the stated model.
- Explain foreign key using the stated model.
- Change one value in the original transfer; explain which assumption still matters.