Exercise walk-through

3.2
SQL queries and transactions

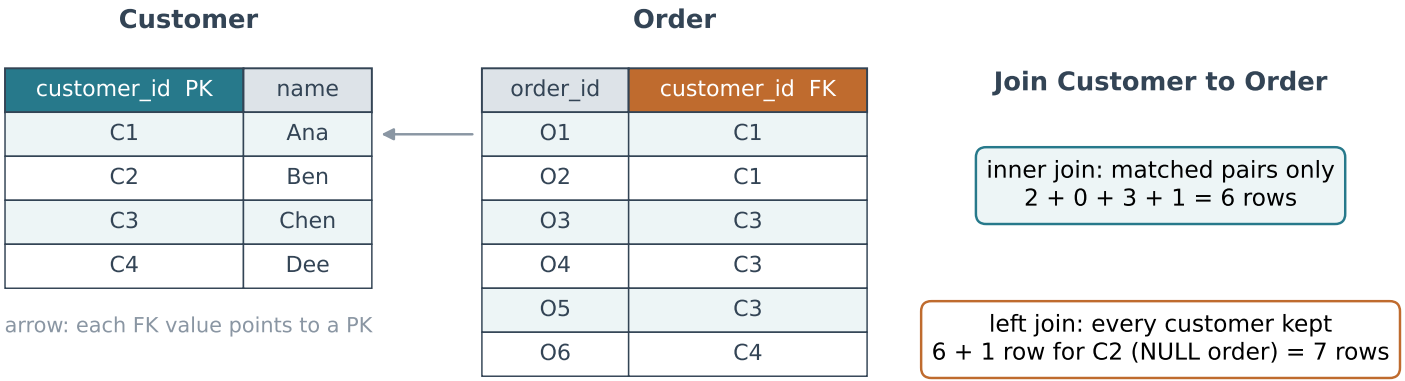
International Baccalaureate · IB Diploma · Computer Science · HL

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 · read the diagram



A foreign key value must match an existing primary key; the join type decides which unmatched rows survive.

- A relational table contains rows and attributes.
- Inner join: 6 matched rows; left join: 7 rows, keeping Ben with empty order fields

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 "
           f"{kind} JOIN o ON c.id = o.cid")
    print(kind, db.execute(sql).fetchone())
db.close()
```

Exercises 2.1·2.3

2.1 Write a query to list the names of students in year 12. [2]

2.2 Write a query to list book titles that start with the letter H, in alphabetical order. [3]

2.3 State the result of `SELECT title FROM Book WHERE book_id <> 11;`. [1]

Exercises 2.1·2.3

2.1 Write a query to list the names of students in year 12. [2]

2.1

- `SELECT name FROM Student`
- `WHERE year_group = 12;`

2.2 Write a query to list book titles that start with the letter H, in alphabetical order. [3]

2.2

- `SELECT title FROM Book`
- `WHERE title LIKE 'H%'`
- `ORDER BY title;`

2.3 State the result of `SELECT title FROM Book WHERE book_id <> 11;`. [1]

2.3

- **Dune, Holes**

Exercises 2.4·3.1

2.4 Write a statement to add the book 13, 'Wonder', by Palacio. [2]

3.1 Which clause filters **groups** after `GROUP BY`? [1]

- ☐ **A** `WHERE`
- **B** `HAVING`
- **C** `ORDER BY`
- **D** `SELECT`

Exercises 2.4·3.1

2.4 Write a statement to add the book 13, 'Wonder', by Palacio. [2]

2.4

- `INSERT INTO Book (book_id, title, author)`
- `VALUES (13, 'Wonder', 'Palacio');`

3.1 Which clause filters **groups** after `GROUP BY`? [1]

- ☐ **A** `WHERE`
- **B** `HAVING`
- **C** `ORDER BY`
- **D** `SELECT`

3.1 B

- `WHERE` filters rows before grouping

Exercises 3.2·3.3

3.2 What does `DELETE FROM Loan;` do? [1]

- ☐ **A** deletes the Loan table's design
- ☐ **B** deletes every row of Loan
- ☐ **C** deletes nothing without `WHERE`
- ☐ **D** deletes the last row

3.3 Which ACID property ensures a committed change survives a power cut? [1]

- ☐ **A** atomicity
- ☐ **B** consistency
- ☐ **C** isolation
- ☐ **D** durability

Exercises 3.2·3.3

3.2 What does `DELETE FROM Loan;` do? [1]

- ☐ **A** deletes the Loan table's design
- ☐ **B** deletes every row of Loan
- ☐ **C** deletes nothing without `WHERE`
- ☐ **D** deletes the last row

3.2 B

- no `WHERE`, so every row

3.3 Which ACID property ensures a committed change survives a power cut? [1]

- ☐ **A** atomicity
- ☐ **B** consistency
- ☐ **C** isolation
- ☐ **D** durability

3.3 D

- durability

Exercise 3.4 (a)

3.4 Use the Student, Book and Loan tables.

(a) Write a query to list the title of every book borrowed by Ana. [4]

Exercise 3.4 (a)

3.4 Use the Student, Book and Loan tables.

(a) Write a query to list the title of every book borrowed by Ana. [4]

(a) One answer:

```
SELECT Book.title
FROM Loan
JOIN Student ON Loan.student_id = Student.student_id
JOIN Book ON Loan.book_id = Book.book_id
WHERE Student.name = 'Ana';
```

- selects `Book.title`
- joins Loan to Student on `student_id`
- joins Loan to Book on `book_id`
- `WHERE` name is `'Ana'`

Exercise 3.4 (b)

3.4 Use the Student, Book and Loan tables.

(b) State the output of your query. [1]

Exercise 3.4 (b)

3.4 Use the Student, Book and Loan tables.

(b) State the output of your query. [1]

(b)

- **Dune, Holes**

Exercise 3.4 (c)

3.4 Use the Student, Book and Loan tables.

(c) Book 10 is returned. Write a statement to remove its loan record. [2]

Exercise 3.4 (c)

3.4 Use the Student, Book and Loan tables.

(c) Book 10 is returned. Write a statement to remove its loan record. [2]

(c)

- `DELETE FROM Loan`
- `WHERE book_id = 10;` (or `loan_id = 100`)

Exercise 3.4 (d)

3.4 Use the Student, Book and Loan tables.

(d) Chen moves to year 12. Write a statement to record this. [2]

Exercise 3.4 (d)

3.4 Use the Student, Book and Loan tables.

(d) Chen moves to year 12. Write a statement to record this. [2]

(d)

- `UPDATE Student SET year_group = 12`
- `WHERE student_id = 3;`

Exercise 3.4 (e)

3.4 Use the Student, Book and Loan tables.

(e) A transfer between two library accounts is processed as a transaction. Explain why atomicity matters. [2]

Exercise 3.4 (e)

3.4 Use the Student, Book and Loan tables.

(e) A transfer between two library accounts is processed as a transaction. Explain why atomicity matters. [2]

(e)

- both the withdrawal and the deposit must happen, or neither
- if the system fails part-way, the transaction is rolled back, so no money is lost or created