```sql
-- Demonstrates aggregation by groups with GROUP BY
-- Uses longlist.db

-- Finds average rating for each book
SELECT "book_id", ROUND(AVG("rating"), 2) AS "average rating" FROM "ratings"
GROUP BY "book_id";

-- Joins titles
SELECT "title", ROUND(AVG("rating"), 2) AS "average rating" FROM "ratings"
JOIN "books" ON "books"."id" = "ratings"."book_id"
GROUP BY "book_id";

-- Chooses books with a rating of 4.0 or higher
SELECT "title", ROUND(AVG("rating"), 2) AS "average rating" FROM "ratings"
JOIN "books" ON "books"."id" = "ratings"."book_id"
GROUP BY "book_id"
HAVING "average rating" > 4.0;
```

```sql
-- Demonstrates joining tables with JOIN
-- Uses sea_lions.db

-- Shows all sea lions for which we have data
SELECT * FROM "sea_lions"
JOIN "migrations" ON "migrations"."id" = "sea_lions"."id";

-- Shows all sea lions, whether or not we have data
SELECT * FROM "sea_lions"
LEFT JOIN "migrations" ON "migrations"."id" = "sea_lions"."id";

-- Shows all data, whether or not there are matching sea lions
SELECT * FROM "sea_lions"
RIGHT JOIN "migrations" ON "migrations"."id" = "sea_lions"."id";

-- Shows all data and all sea lions
SELECT * FROM "sea_lions"
FULL JOIN "migrations" ON "migrations"."id" = "sea_lions"."id";

-- JOINs sea lions and migrations without specifying matching column
SELECT * FROM "sea_lions"
NATURAL JOIN "migrations";

-- Uses WHERE after joining a table
SELECT * FROM "sea_lions"
JOIN "migrations" ON "migrations"."id" = "sea_lions"."id"
WHERE "migrations"."distance" > 1500;
```

```sql
-- Demonstrates subqueries
-- Uses longlist.db

-- Finds all books published by MacLehose Press, with hard-coded id
SELECT "id" FROM "publishers" WHERE "publisher" = 'MacLehose Press';

SELECT "title" FROM "books" WHERE "publisher_id" = 12;

-- Finds all books published by MacLehose Press, with a nested query
SELECT "title" FROM "books" WHERE "publisher_id" = (
    SELECT "id" FROM "publishers" WHERE "publisher" = 'MacLehose Press'
);

-- Finds all ratings for "In Memory of Memory"
SELECT "rating" FROM "ratings" WHERE "book_id" = (
    SELECT "id" FROM "books" WHERE "title" = 'In Memory of Memory'
);

-- Finds average rating for "In Memory of Memory"
SELECT AVG("rating") FROM "ratings" WHERE "book_id" = (
    SELECT "id" FROM "books" WHERE "title" = 'In Memory of Memory'
);

-- Finds author who wrote "The Birthday Party"
SELECT "id" FROM "books" WHERE "title" = 'The Birthday Party';

SELECT "author_id" FROM "authored" WHERE "book_id" = (
    SELECT "id" FROM "books" WHERE "title" = 'The Birthday Party'
);

SELECT "name" FROM "authors" WHERE "id" = (
    SELECT "author_id" FROM "authored" WHERE "book_id" = (
        SELECT "id" FROM "books" WHERE "title" = 'The Birthday Party'
    )
);

-- Finds all books by Fernanda Melchor, using IN
SELECT "id" FROM "authors" WHERE "name" = 'Fernanda Melchor';

SELECT "book_id" FROM "authored" WHERE "author_id" = (
    SELECT "id" FROM "authors" WHERE "name" = 'Fernanda Melchor'
);
```

```sql

SELECT "title" FROM "books" WHERE "id" IN (
    SELECT "book_id" FROM "authored" WHERE "author_id" = (
        SELECT "id" FROM "authors" WHERE "name" = 'Fernanda Melchor'
    )
);

-- Uses IN to search for multiple authors
SELECT "title" FROM "books" WHERE "id" IN (
    SELECT "book_id" FROM "authored" WHERE "author_id" IN (
        SELECT "id" FROM "authors" WHERE "name" IN ('Fernanda Melchor', 'Annie Ernaux')
    )
);
```

```sql
-- Demonstrates set operations
-- Uses longlist.db

-- UNION
-- Selects all authors, labeling as authors
SELECT 'author' AS "profession", "name" FROM "authors";

-- Selects all translators, labeling as translators
SELECT 'translator' AS "profession", "name" FROM "translators";

-- Combines authors and translators into one result set
SELECT 'author' AS "profession", "name" FROM "authors";
UNION
SELECT 'translator' AS "profession", "name" FROM "translators";

-- INTERSECT (Assume names are unique)
-- Finds authors and translators
SELECT "name" FROM "authors"
INTERSECT
SELECT "name" FROM "translators";

-- Finds books translated by Sophie Hughes
SELECT "book_id" FROM "translated" WHERE "translator_id" = (
    SELECT "id" FROM "translators" WHERE name = 'Sophie Hughes'
);

-- Finds books translated by Margaret Jull Costa
SELECT "book_id" FROM "translated" WHERE "translator_id" = (
    SELECT "id" FROM "translators" WHERE name = 'Margaret Jull Costa'
);

-- Finds intersection of books
SELECT "book_id" FROM "translated" WHERE "translator_id" = (
    SELECT "id" FROM "translators" WHERE name = 'Sophie Hughes'
)
INTERSECT
SELECT "book_id" FROM "translated" WHERE "translator_id" = (
    SELECT "id" FROM "translators" WHERE name = 'Margaret Jull Costa'
);

-- Finds intersection of books
SELECT "title" FROM "books" WHERE "id" = (
    SELECT "book_id" FROM "translated" WHERE "translator_id" = (
    SELECT "id" FROM "translators" WHERE name = 'Sophie Hughes'
    )
    INTERSECT
    SELECT "book_id" FROM "translated" WHERE "translator_id" = (
        SELECT "id" FROM "translators" WHERE name = 'Margaret Jull Costa'
    )
);

-- EXCEPT (Assume names are unique)
-- Finds translators who are not authors
SELECT "name" FROM "translators"
EXCEPT
SELECT "name" FROM "authors";
```