<k/>

SQL BASICS

Read, filter, join and summarise data

THEKHANSA.COM

READ DATA

```
SELECT name, city
FROM customers
WHERE city = 'London'
ORDER BY name ASC
LIMIT 10;

SELECT DISTINCT city FROM customers;
```

FILTERS

```
WHERE age >= 18 AND city = 'Paris'
WHERE city IN ('London', 'Berlin')
WHERE amount BETWEEN 10 AND 50
WHERE name LIKE 'A%'     -- starts with A
WHERE email IS NULL
WHERE NOT (age < 18)
```

SUMMARIES

```
COUNT(*)  SUM(x)  AVG(x)  MIN(x)  MAX(x)

SELECT city, COUNT(*) AS n
FROM customers
GROUP BY city
HAVING COUNT(*) > 5
ORDER BY n DESC;
```

JOINS

```
SELECT o.id, c.name
FROM orders o
JOIN customers c ON c.id = o.customer_id;

-- INNER JOIN: only matches
-- LEFT JOIN:  all from the left table
-- RIGHT JOIN: all from the right table
-- FULL JOIN:  everything from both
```

CHANGE DATA

```
INSERT INTO t (a, b) VALUES (1, 'x');
UPDATE t SET b = 'y' WHERE a = 1;
DELETE FROM t WHERE a = 1;

CREATE TABLE t (
  id INT PRIMARY KEY,
  name TEXT NOT NULL
);
```

CASE AND NULLS

```
SELECT name,
  CASE WHEN score >= 50 THEN 'pass'
       ELSE 'fail' END AS result
FROM exams;

COALESCE(x, 0)   -- first non-NULL
NULL = NULL is not true: use IS NULL
```

SUBQUERIES AND CTES

```
WITH big AS (
  SELECT customer_id, SUM(amount) AS total
  FROM orders GROUP BY customer_id
)
SELECT c.name, big.total
FROM big JOIN customers c
  ON c.id = big.customer_id
WHERE big.total > 1000;
```

WINDOW FUNCTIONS

```
SELECT name, dept, salary,
  RANK() OVER (
    PARTITION BY dept
    ORDER BY salary DESC) AS rnk,
  SUM(salary) OVER (PARTITION BY dept)
    AS dept_total
FROM staff;
-- ROW_NUMBER, RANK, LAG, LEAD
```

COMBINE RESULTS

```
SELECT a FROM t1
UNION          -- removes duplicates
SELECT a FROM t2;

UNION ALL      -- keeps duplicates
INTERSECT   EXCEPT   -- support varies
```

Good to know: SQL runs in this order: FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY, LIMIT. That is why you cannot use a SELECT alias inside WHERE. Date functions and LIMIT differ between databases.