常見 SQL 指令總表

以下整理自簡報內容:contentReference[oaicite:0]index=0，分為資料定義、資料操作與進階查詢三大類。

1. 資料定義 (DDL)

- CREATE：建立資料庫物件

```
CREATE TABLE EMPLOYEE (
  id VARCHAR(15) PRIMARY KEY,
  name VARCHAR(50) NOT NULL,
  birthday DATE NOT NULL
);
```

- ALTER：修改資料表

```
ALTER TABLE EMPLOYEE ADD job VARCHAR(20);
ALTER TABLE EMPLOYEE DROP COLUMN birthday CASCADE;
```

- DROP：刪除資料表或 schema

```
DROP TABLE EMPLOYEE CASCADE;
DROP SCHEMA RETAIL CASCADE;
```

2. 基本查詢 (SELECT)

- **基本語法**

```
SELECT <屬性清單>
FROM <資料表>
WHERE <條件>;
```

- **條件與運算子**

```
-- 等於 / 不等於
SELECT * FROM EMPLOYEE WHERE gender = 'M';
SELECT * FROM EMPLOYEE WHERE id <> 'A123';

-- LIKE 與萬用字元
SELECT name FROM EMPLOYEE WHERE name LIKE 'Chi%';

-- NULL 判斷
SELECT name FROM EMPLOYEE WHERE supervisor_id IS NULL;
```

- **集合運算**

```sql
SELECT salary FROM EMPLOYEE WHERE store_id = 1
UNION
SELECT salary FROM EMPLOYEE WHERE store_id = 2;
```

- **JOIN**

```sql
-- 內部連接
SELECT e.name, s.postal_code
FROM EMPLOYEE e
JOIN STORE s ON e.store_id = s.id;

-- 左外連接
SELECT e.name, s.mgr_id
FROM EMPLOYEE e
LEFT JOIN STORE s ON e.id = s.mgr_id;
```

- **排序與聚合**

```sql
SELECT store_id, COUNT(*), AVG(monthly_salary)
FROM EMPLOYEE
GROUP BY store_id
HAVING COUNT(*) > 2
ORDER BY AVG(monthly_salary) DESC;
```

3. 資料操作 (DML)

- **INSERT**

```sql
INSERT INTO EMPLOYEE(id, name, gender, birthday, monthly_salary)
VALUES('A123456789', 'Xiao-Ming Wang', 'M', '1999-01-01', 120000);
```

- **DELETE**

```sql
DELETE FROM EMPLOYEE WHERE name = 'Chih-Yuan Lee';
```

- **UPDATE**

```sql
UPDATE EMPLOYEE
SET monthly_salary = monthly_salary * 1.1
WHERE store_id = 1;
```

4. 進階查詢

- **子查詢與集合比較**

```sql
SELECT name
FROM EMPLOYEE
WHERE monthly_salary > ALL (
  SELECT monthly_salary FROM EMPLOYEE WHERE store_id = 1
);
```

- ## LIMIT & OFFSET

```sql
SELECT name FROM EMPLOYEE
ORDER BY birthday ASC
LIMIT 3 OFFSET 1;
```

- ## CASE 條件判斷

```sql
SELECT id, name,
CASE SUBSTRING(id,1,1)
  WHEN 'A' THEN 'Taipei'
  WHEN 'S' THEN 'Kaohsiung'
  ELSE 'Others'
END AS region
FROM EMPLOYEE;
```

- ## CTE (Common Table Expression)

```sql
WITH MaxSalary AS (
  SELECT MAX(monthly_salary) AS max_sal
  FROM EMPLOYEE WHERE store_id = 2
)
SELECT name FROM EMPLOYEE
WHERE monthly_salary > (SELECT max_sal FROM MaxSalary);
```

- ## VIEW

```sql
CREATE VIEW STORE_SALES_VIEW AS
SELECT s.store_id, sd.product_id, sd.qty
FROM STORE s
JOIN SALES_DETAIL sd ON s.id = sd.store_id;
```