VIEWS

Subqueries and joins become complex. This is where views come to rescue.

```
SELECT C.CLIENT_ID, C.NAME, SUM(INVOICE_TOTAL) AS TOTAL_SALES
FROM CLIENTS C
JOIN INVOICES I
USING(CLIENT_ID)
GROUP BY C.CLIENT_ID, C.NAME;
```

Instead of writing select query each time, we can save this query in a view and use that view in many places.

```
CREATE VIEW SALES_BY_CLIENT AS
SELECT C.CLIENT_ID, C.NAME, SUM(INVOICE_TOTAL) AS TOTAL_SALES
FROM CLIENTS C
JOIN INVOICES I
USING(CLIENT_ID)
GROUP BY C.CLIENT_ID, C.NAME;
```

We can use this view just like table.

```
SELECT * FROM SALES_BY_CLIENT;

SELECT * FROM SALES_BY_CLIENT
ORDER BY TOTAL_SALES DESC;

SELECT * FROM SALES_BY_CLIENT
WHERE TOTAL_SALES > 5000;

SELECT * FROM SALES_BY_CLIENT
JOIN CLIENTS
USING(CLIENT_ID);
```

Views act as virtual table. Views don’t store data. Data is stored in a table. A view just provides a view to a underlying table.