VIEWS

VIEWS ARE 'VIRTUAL TABLES'

LET'S SAY THERE IS A REALLY COMPLICATED QUERY YOU USE AGAIN AND AGAIN

YOU COULD DEFINE A VIEW THAT MAPS TO THAT QUERY

LET'S SAY THERE IS A REALLY COMPLICATED QUERY YOU USE AGAIN AND AGAIN

YOU COULD DEFINE A VIEW THAT MAPS TO THAT QUERY

YOU COULD THEN QUERY THAT VIEW EXACTLY AS IF IT WERE A TABLE

RATHER THAN HAVE IT AS A COMPLICATED SUB-QUERY IN YOUR OTHER QUERIES

LET'S SAY THERE IS A REALLY COMPLICATED QUERY YOU USE AGAIN AND AGAIN

YOU COULD DEFINE A VIEW THAT MAPS TO THAT QUERY

YOU COULD THEN QUERY THAT VIEW EXACTLY AS IF IT WERE A TABLE

YOU CAN'T UPDATE OR INSERT OR DELETE DATA IN A VIEW QUITE AS EASILY AS A REAL TABLE THOUGH

LET'S SAY THERE IS A REALLY COMPLICATED QUERY YOU USE AGAIN AND AGAIN

YOU COULD DEFINE A VIEW THAT MAPS TO THAT QUERY

YOU COULD THEN QUERY THAT VIEW EXACTLY AS IF IT WERE A TABLE

YOU CAN'T UPDATE OR INSERT OR DELETE DATA IN A VIEW QUITE AS EASILY AS A REAL TABLE THOUGH

WRITING TO A VIEW IS COMPLICATED AND WE WON'T DISCUSS THIS IN A LOT OF DETAIL

WRITING TO A VIEW IS COMPLICATED AND WE WON'T DISCUSS THIS IN A LOT OF DETAIL

A VIEW CAN BE UPDATED IF 4 CONDITIONS ARE MET:

- THE FROM CLAUSE HAS ONLY 1 RELATION
- THE SELECT CLAUSE CONTAINS ONLY COLUMN NAMES, NOT AGGREGATE OPERATORS, DISTINCT OR EXPRESSIONS
- ANY COLUMN NOT INCLUDED IN THE SELECT CLAUSE ACCEPTS NULLS, OR HAS A DEFAULT VALUE
- THE QUERY HAS NO GROUP BY OR HAVING CLAUSE

LET'S SAY THERE IS A REALLY COMPLICATED QUERY YOU USE AGAIN AND AGAIN

YOU COULD DEFINE A VIEW THAT MAPS TO THAT QUERY

```
CREATE VIEW Top_Selling_Products AS
SELECT
      P.PRODUCTNAME,YEAR(DATE),SUM(REVENUE)
FROM
     (SELECT * FROM SALES_DATA WHERE PRODUCTID IN TOP_SELLERS)
INNER JOIN
     PRODUCTS P
     ON
     S.PRODUCTID = P.PRODUCTID
WHERE
     (P.PRODUCTNAME IN (SELECT PRODUCTNAME FROM TOP_SELLERS))
    AND
     (YEAR(DATE) = (SELECT YEAR(MAX(DATE)) FROM SALES_DATA)
```

DEFINE THEM EXACTLY AS YOU WOULD TABLES - JUST SAY VIEW INSTEAD OF TABLE

TEMPORARY TABLES

TEMPORARY TABLES ARE TABLES THAT LAST ONLY FOR THE DURATION OF A USER SESSION

```
CREATE TEMPORARY TABLE T_Selling_Products AS
SELECT
      P.PRODUCTNAME,YEAR(DATE),SUM(REVENUE)
FROM
     (SELECT * FROM SALES_DATA WHERE PRODUCTID IN TOP_SELLERS)
INNER JOIN
     PRODUCTS P
     ON
     S.PRODUCTID = P.PRODUCTID
WHERE
     (P.PRODUCTNAME IN (SELECT PRODUCTNAME FROM TOP_SELLERS))
    AND
     (YEAR(DATE) = (SELECT YEAR(MAX(DATE)) FROM SALES_DATA)
```

TEMPORARY TABLES ARE TABLES THAT LAST ONLY FOR THE DURATION OF A USER SESSION

TEMP TABLES ARE YOUR FRIENDS - USE THEM, PARTICULARLY IF YOU DELETE DATA OR DROP TABLES. BACK UP THE OLD TABLE FIRST

GRANT AND REVOKE

GRANT AND REVOKE ARE SQL COMMANDS TO CONTROL USER PRIVILEGES

```
GRANT SELECT,INSERT,DELETE ON Products TO Vitthal
REVOKE SELECT,INSERT,DELETE ON Products TO Vitthal
```

THERE IS RICH FUNCTIONALITY AROUND USER PRIVILEGES IN SQL - READ FURTHER IF THIS IS INTERESTING TO YOU