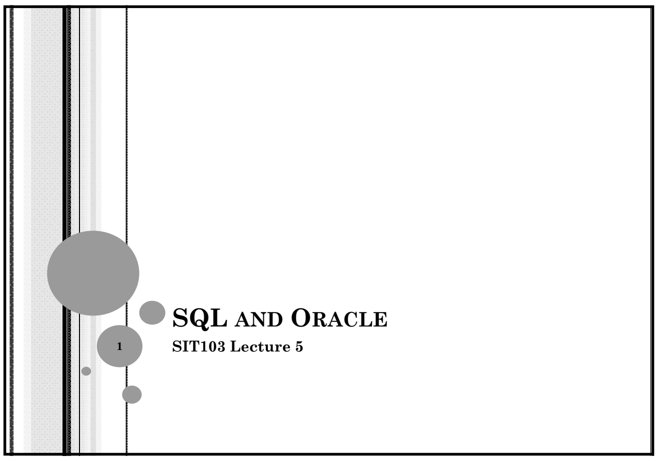
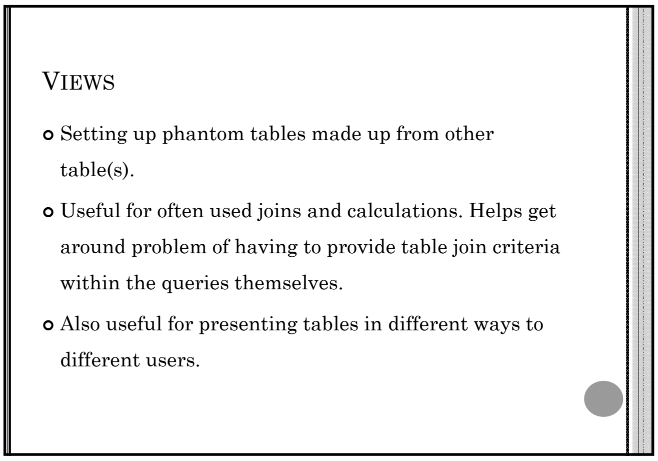


SQL and Oracle

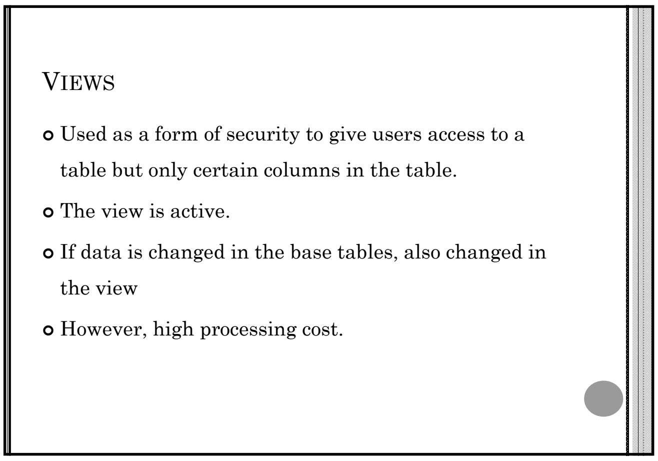
SIT103 Lecture 5

1



Views

- Setting up phantom tables made up from other table(s).
- Useful for often used joins and calculations. Helps get around problem of having to provide table join criteria within the queries themselves.
- Also useful for presenting tables in different ways to different users.



Views

- Used as a form of security to give users access to a table but only certain columns in the table.
- The view is active.
- If data is changed in the base tables, also changed in the view
- However, high processing cost.

VIEWS

-
  ```
  CREATE OR REPLACE VIEW COMPSTUD AS
        SELECT *
        FROM STUDENT
        WHERE PROGRAMME_CODE LIKE '100%';
  ```
- No physical table is created; No records are copied;The query ONLY is stored
- Can run query like this, using above VIEW:

  ```
        SELECT SURNAME, GIVEN
        FROM COMPSTUD
        WHERE SURNAME = 'SMITH';
  ```

VIEWS

```
CREATE VIEW STUDPGM AS
   SELECT STUDENT_NO, GIVEN || ' ' || SURNAME AS
   NAME,        PROGRAMME.PROGRAMME_CODE AS PC,
   PROGRAMME_NAME
   FROM STUDENT S, PROGRAMME P
   WHERE S.PROGRAMME_CODE = P.PROGRAMME_CODE;
```

- Can run query like this, using above VIEW:

  ```
        SELECT *
        FROM STUDPGM
        WHERE PC = 100;
  ```

LAYERED VIEWS

- ◆ Once a view is created, it can be used just like any other table
- ◆ A view can therefore use another view
- ◆ This technique can be used to layer complex queries into multiple levels

```
CREATE VIEW FEMALECOMPSTUD AS
SELECT *
FROM COMPSTUD
WHERE SEX = 'F';
```

Views

- To remove a view (deactivate it)

```
DROP  VIEW  STUDPGM;
```

Combining Select Statements

- Column results of two select statements are combined into one result set.
- **Union**: Returns only distinct rows that appear in either result
- **Intersect:**Returns only those rows returned by both queries
- **Minus**: Returns only unique rows returned by the first query but not by the second

Combining Select Statements

UNION: Returns only distinct rows that appear in either result

```
SELECT GIVEN, DOB
        FROM STUDENT
        WHERE SURNAME = 'SMITH'
        UNION
        SELECT GIVEN, DOB
        FROM STAFF
        WHERE SURNAME = 'SMITH'
```

COMBINING SELECTS – UNION QUERY

STUDENT	
STUD_NO	**NAME**
8901234Z	Jeffrey
9004567A	Angela
9103876R	Wilma
9901123S	John
9901126T	Peter

STAFF		
STAFF_NO	**NAME**	**SEX**
10023	John	M
10025	Harry	M
10026	Jeff	M
10032	Peter	F

SELECT STUDENT_NO, GIVEN,'', 'STUDENT' FROM STUDENT

UNION

SELECT STAFFNO, GIVEN, SEX, 'STAFF' FROM STAFF;

COMBINING SELECTS – UNION RESULT

STUDENT		**STAFF**	
STUD_NO	**NAME**	**SEX**	**COLUMN??**
8901234Z	Jeffrey		STUDENT
9004567A	Angela		STUDENT
9103876R	Wilma		STUDENT
9901123S	John		STUDENT
9901126T	Peter		STUDENT
10023	John	M	STAFF
10025	Harry	M	STAFF
10026	Jeff	M	STAFF
10032	Peter	F	STAFF

COMBINING SELECT STATEMENTS

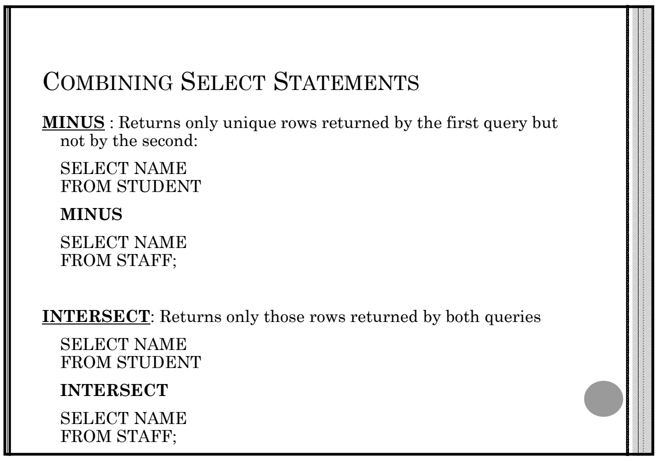
- Columns of each select must match by type & order placed in SELECT clause.
- ORDER BY cannot be specified on each select clause but can be specified on whole result set.

SELECT NAME
FROM STUDENT

UNION

SELECT NAME
FROM STAFF

ORDER BY NAME



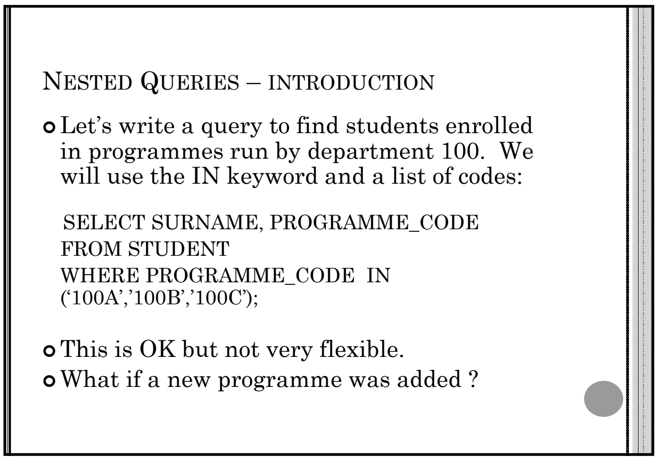
COMBINING SELECT STATEMENTS

MINUS : Returns only unique rows returned by the first query but not by the second:

```
SELECT NAME
FROM STUDENT
MINUS
SELECT NAME
FROM STAFF;
```

INTERSECT: Returns only those rows returned by both queries

```
SELECT NAME
FROM STUDENT
INTERSECT
SELECT NAME
FROM STAFF;
```

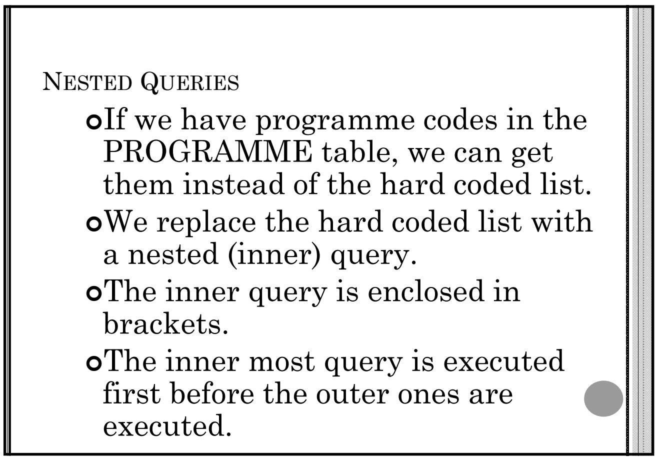


NESTED QUERIES – INTRODUCTION

- Let's write a query to find students enrolled in programmes run by department 100. We will use the IN keyword and a list of codes:

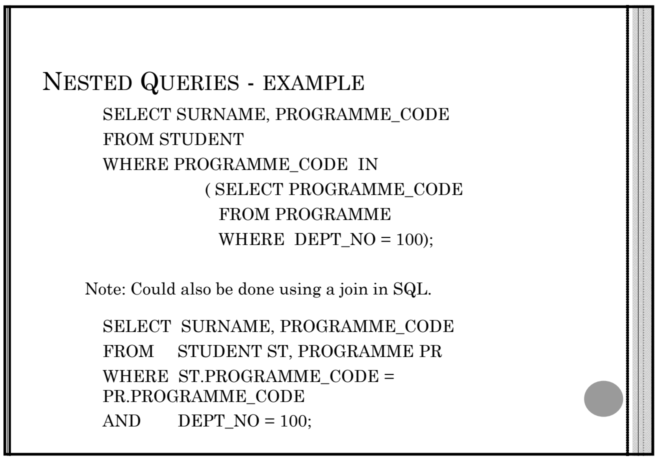
```
SELECT SURNAME, PROGRAMME_CODE
FROM STUDENT
WHERE PROGRAMME_CODE  IN
('100A','100B','100C');
```

- This is OK but not very flexible.
- What if a new programme was added ?



NESTED QUERIES

- If we have programme codes in the PROGRAMME table, we can get them instead of the hard coded list.
- We replace the hard coded list with a nested (inner) query.
- The inner query is enclosed in brackets.
- The inner most query is executed first before the outer ones are executed.

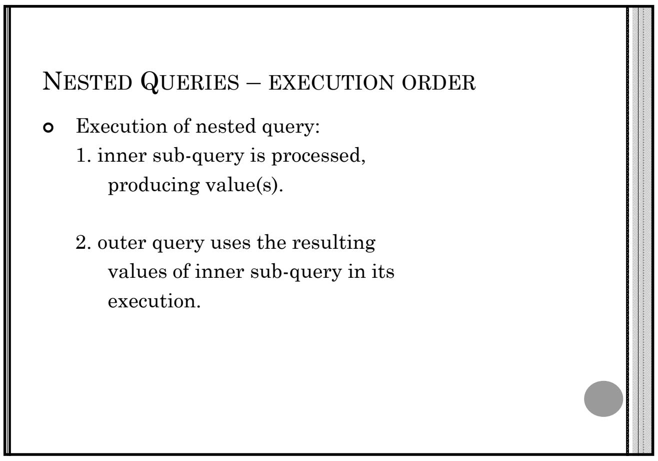


Nested Queries - example

```
SELECT SURNAME, PROGRAMME_CODE
FROM STUDENT
WHERE PROGRAMME_CODE  IN
              ( SELECT PROGRAMME_CODE
                FROM PROGRAMME
                WHERE  DEPT_NO = 100);
```

Note: Could also be done using a join in SQL.

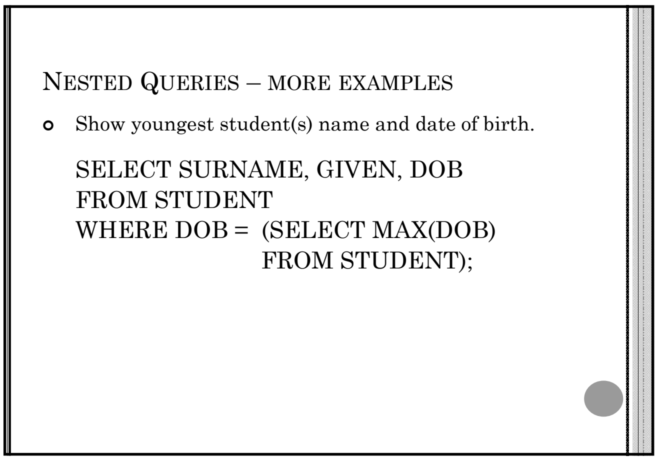
```
SELECT  SURNAME, PROGRAMME_CODE
FROM     STUDENT ST, PROGRAMME PR
WHERE  ST.PROGRAMME_CODE =
PR.PROGRAMME_CODE
AND       DEPT_NO = 100;
```



Nested Queries – execution order

- Execution of nested query:
  1. inner sub-query is processed, producing value(s).

  2. outer query uses the resulting values of inner sub-query in its execution.



Nested Queries – more examples

- Show youngest student(s) name and date of birth.

```
SELECT SURNAME, GIVEN, DOB
FROM STUDENT
WHERE DOB =  (SELECT MAX(DOB)
                        FROM STUDENT);
```

Nested Queries – more examples

- Show all programmes that do not have students enrolled in them.

```
SELECT PROGRAMME_NAME
FROM    PROGRAMME
WHERE PROGRAMME_CODE NOT IN
      (SELECT DISTINCT
PROGRAMME_CODE
       FROM STUDENT);
```

Finding the Largest of a Summary

- Revisit the Group By query again

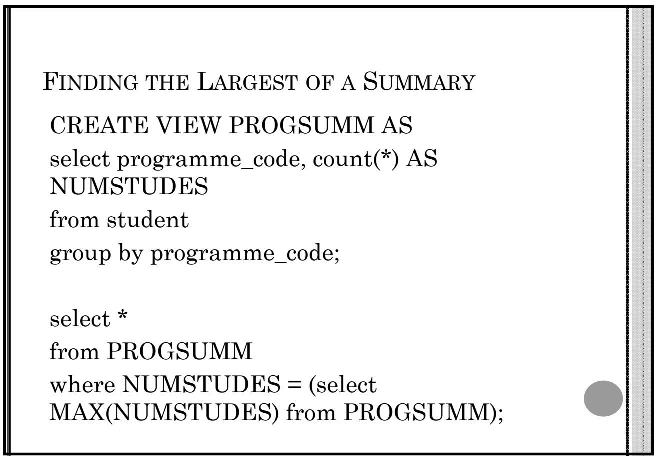
```
SELECT PROGRAMME_CODE, COUNT(*)

FROM STUDENT
GROUP BY PROGRAMME_CODE ;
```

- Want to find the programme with the highest number of students

Finding the Largest of a Summary

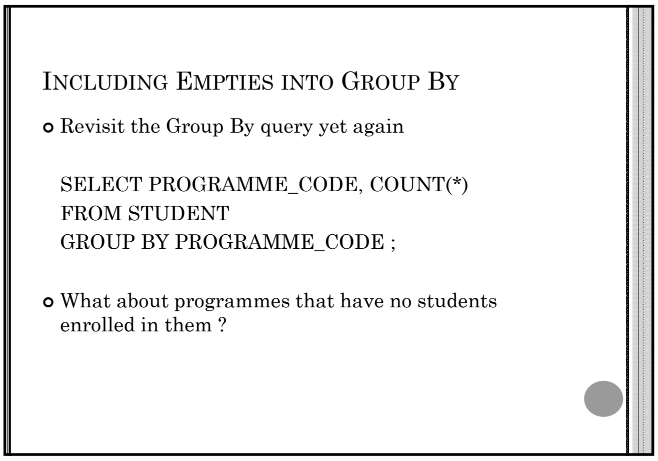
- Hard to combine the group by and a nested query in one query
- Make the Group By query into a view
- Then use a nested query on the view to find the highest count



Finding the Largest of a Summary

```
CREATE VIEW PROGSUMM AS
select programme_code, count(*) AS
NUMSTUDES
from student
group by programme_code;

select *
from PROGSUMM
where NUMSTUDES = (select
MAX(NUMSTUDES) from PROGSUMM);
```

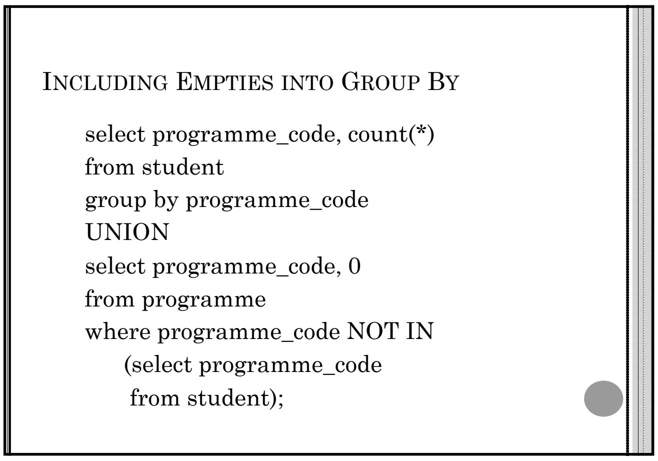


Including Empties into Group By

- Revisit the Group By query yet again

```
SELECT PROGRAMME_CODE, COUNT(*)
FROM STUDENT
GROUP BY PROGRAMME_CODE ;
```

- What about programmes that have no students enrolled in them ?



Including Empties into Group By

```
select programme_code, count(*)
from student
group by programme_code
UNION
select programme_code, 0
from programme
where programme_code NOT IN
      (select programme_code
       from student);
```