MySQL Unit 09 – Subqueries, Unions, & Views

**\*\*\*ONLY SELECT CODE YOU WANT TO EXECTUE\*\*\***

**SUBQUERIES:**

Embedding a SELECT statement inside another SELECT statement (usually inside the WHERE clause)

WHERE Salary > (SELECT AVG(Salary) from Faculty)

\*\*Typically subqueries return a single row/ single value 🡪 this returns the AVG salary

**UNIONS:**

SELECT Name FROM Department ORDER BY Name;

SELECT Name FROM MAJOR ORDER BY Name;

\*\*When you want the result set to be the two names stacked together/ concatenated together

- Insert the UNION keyword:

SELECT Name FROM Department ODER BY Name;

UNION

SELECT Name FROM MAJOR ODER BY Name;

\*\*Usually its necessary to hard code column names to distinguish between the two tables info:

SELECT ‘Department’ AS Area,

Name FROM Department 🡪 No need to ORDER BY in both statements

UNION

SELECT ‘Major’ AS Area,

Name FROM MAJOR ORDER BY Name;

**\*\*\*ONLY SELECT CODE YOU WANT TO EXECTUE\*\*\***

**CREATING VIEWS:**

A view is a SELECT statement that is stored

A view is a virtual table

The results of a view are alive only when you run the view; the data returned is never actually stored

* Therefore the data is always up to date

\*\*Often convenient to use views to knit together information & then store it as a virtual table in the DB

When you create a view, you can then use it in the SELECT statement the same way we’ve been using tables

**STACKING VIEWS:**

CREATE OR REPLACE VIEW StudentFlattened AS 🡪 creates the view

//Code you want to reuse; 🡪 uses the view

SELECT Statement to use the view;

\*\*‘Flattened’ takes two tables where data is broken up and flattened it into one view

CREATE OR REPLACE VIEW StudentsWithoutAScholarship AS

SELECT Name, Email, Major

FROM StudentsFlattened 🡪 reuses the existing view (Stacks the views)

WHERE Scholarship = 0;

SELECT Name, Email 🡪 uses the view

FROM StudentsWithoutAScholarship;