## Queries

1. query to find people who are absent in each course on a *particular date* and display course\_id,name,finger\_id,course\_name (FOR SENDING MSG TO ABSENTEES) NOTE: Change the date clause depending on the current system date, currently initialized a default value

SELECT \* FROM (SELECT DISTINCT course\_id,finger\_id FROM (SELECT course\_id,finger\_id FROM (SELECT course\_id,finger\_id FROM (SELECT fiSELECT \* FROM (SELECT DISTINCT course\_id,finger\_id FROM (SELECT course\_id,finger\_id FROM (SELECT course\_id,finger\_id FROM (SELECT course\_id,attendance,cast(timestamp as date) as date FROM course\_attendance) AS a WHERE date='2018-04-06') AS b NATURAL JOIN (SELECT \* FROM course\_taken)AS c ) AS x WHERE NOT EXISTS (SELECT DISTINCT course\_id,finger\_id FROM (SELECT course\_id,finger\_id FROM (SELECT finger\_id,course\_id,attendance,cast(timestamp as date) as date FROM course\_attendance) AS d WHERE date='2018-04-06') AS e WHERE x.course\_id=course\_id AND x.finger\_id=finger\_id)) AS y NATURAL JOIN (SELECT name,s\_id,finger\_id FROM student) AS z NATURAL JOIN (SELECT course\_id,course\_name FROM faculty) AS w

2.This shows that attendance of student out of total lecs for a course (FOR STUDENT PAGE) NOTE: accept username from app that will be the id for which this info has to be displayed. So, a where clause has to be added

SELECT \* FROM (SELECT course\_id,course\_name,finger\_id, total\_attendance,lecs FROM (SELECT course\_id,finger\_id,COUNT(attendance) as total\_attendance FROM course\_attendance GROUP BY course\_id,finger\_id) AS a NATURAL JOIN (SELECT \* FROM faculty) AS b) AS k NATURAL JOIN (SELECT name,s\_id,finger\_id FROM student) AS l

3. These are the students whose entry is not present for a particular course that means the attendance is 0 for that course out of the total lecs  $(FOR\ STUDENT\ PAGE\ )$ 

SELECT\* FROM (SELECT \* FROM (SELECT course\_id,finger\_id FROM course\_taken AS x WHERE NOT EXISTS(SELECT course\_id,course\_name,finger\_id, total\_attendance,lecs FROM (SELECT course\_id,finger\_id,COUNT(attendance) as total\_attendance FROM course\_attendance GROUP BY course\_id,finger\_id) AS a NATURAL JOIN (SELECT \* FROM faculty) AS b WHERE x.course\_id=course\_id and x.finger\_id=finger\_id)) AS y NATURAL JOIN (SELECT course\_name,course\_id,lecs FROM faculty) AS z) AS k NATURAL JOIN (SELECT name,s id,finger id FROM student) AS l

4.Shows date wise attendance in the course to faculty (FOR FACULTY page) NOTE: to be displayed depending on username of faculty fetched from app so a where clause needs to be added for that

## **SELECT \* FROM (SELECT**

course\_id,course\_name,f\_id,name,total\_attendance,students,date FROM (SELECT course\_id,cast(timestamp as date) as date,COUNT(attendance) as total\_attendance FROM course\_attendance GROUP BY course\_id,cast(timestamp as date)) AS a NATURAL JOIN (SELECT \* FROM faculty) AS b) AS k

5. Trigger - Now when attendance is taken for a course in faculty table the lecs should be incremented corresponding to that course id (Note: has been already added, no need to write php for this)

CREATE DEFINER='id5262205\_riya'@'%' TRIGGER 'updateLecs' BEFORE INSERT ON 'course\_attendance' FOR EACH ROW BEGIN IF (cast(NEW.timestamp as date) Not IN (SELECT DISTINCT cast(timestamp as date) FROM course\_attendance)) THEN UPDATE faculty SET lecs = lecs + 1 WHERE course\_id = NEW.course\_id; END IF; END