

## 1.Insert

```
MODEL FOR
`INSERT INTO REQUESTSORCOMPLAINTS
(SENDER_TYPE, SENDER_ID, SENDING_TIME, MESSAGE_TYPE, MESSAGE, MESSAGE_ID)
VALUES('${senderType}', '${id}', SYSDATE, 'Complaint', '${dataReceived.Complaint}', 'MSG' || MSG_ID.NEXTVAL)`
;
```

Insert into RequestOrComplaints Table from student or teacher

```
INSERT INTO Enrolls (Student_ID,Session_Name, Course_ID, Status)
VALUES ('${id}', '${session_name}', '${course_id}','${status}')
```

Insert into enrolls from Student add or drop courses

```
insert into STUDENT (Student_ID, First_Name, Last_Name, Level_Term, Department, Email, Phone_No,
Nid_No, Address, Credits_Completed, Section_name, Advisor_ID, Password, img_url)
values ('${dataReceived.STUDENT_ID}', '${dataReceived.FIRST_NAME}', '${dataReceived.LAST_NAME}',
'${dataReceived.LEVEL_TERM}', '${dataReceived.DEPARTMENT}', '${dataReceived.EMAIL}',
'${dataReceived.PHONE_NO}', '${dataReceived.NID_NO}', '${dataReceived.ADDRESS}', ${dataReceived.CREDITS_COMPLETED},
'${dataReceived.SECTION_NAME}', '${dataReceived.ADVISOR_ID}', '${hashedPW}', 'https://placeholder.jp/100x100.png');
END
```

Insert into Student From add student by the admin

```
`INSERT INTO Exam(Course_ID, Session_Name, Exam_Date)
VALUES ('${courseID}', '${sessionName}', TO_DATE('${date}', 'YYYY-MM-DD'))`
```

Insert into Exam Table by Admin

```
`INSERT INTO PROJECTDBA.RESULTS
(STUDENT_ID, COURSE_ID, ATTENDANCE, CT1, CT2, CT3, CT4, CT5, TERM_FINAL, TOTAL, GRADE)
VALUES('${dataReceived["STUDENT_ID"]}', '${dataReceived["COURSE_ID"]}', ${
arr2[0]
},
${arr2[1]}, ${arr2[2]}, ${arr2[3]}, ${arr2[4]}, ${Number(
dataReceived["Attendance"]
)},
${Number(dataReceived["Term Final"])}, ${total}, ${grade})`
;
```

Insert Into Result table by a teacher

```

await run(
  DECLARE
    sessName varchar2(30);
  BEGIN
    public_package.logged_in_id := '${id}';
    Insert into Courses (Course_ID, Course_Title, Department, Credit_Hours, Level_Term, Department_Offered, Prerequisites)
    values (
      '${dataReceived.COURSE_ID}',
      '${dataReceived.COURSE_TITLE}',
      '${dataReceived.DEPARTMENT}',
      '${dataReceived.CREDIT_HOURS}',
      '${dataReceived.LEVEL_TERM}',
      '${dataReceived.DEPARTMENT_OFFERED}',
      '${dataReceived.PREREQUISITES}'
    );

    IF '${currentSession}' LIKE 'Jul%' and '${dataReceived.LEVEL_TERM}' like '%-2' then
      sessName := '${currentSession}';
    elsif '${currentSession}' LIKE 'Jan%' and '${dataReceived.LEVEL_TERM}' like '%-1' then
      sessName := '${currentSession}';
    else
      nameNextSession('${currentSession}', sessName);
    end if;

    INSERT INTO PROJECTDBA.SESSIONS
    (SESSION_NAME, COURSE_ID)
    VALUES(sessName, '${dataReceived.COURSE_ID}');

  END;
)

```

Insert into courses by admin, also insert in session

## 2.Update

```

await run(`UPDATE Enrolls
SET Status = 'Current'
WHERE Student_ID LIKE '${studentID}' AND COURSE_ID LIKE '${course_ID}' AND Status LIKE 'Pending'`);

```

Admin sets a student's 'pending' course status to 'current' status by approving

```

UPDATE PROJECTDBA.TEACHER
SET FIRST_NAME='${req.body.first_name}', LAST_NAME='${req.body.last_name}', EMAIL='${req.body.email}', PHONE_NO='${req.body.phone_no}', ADDRESS='${req.

```

Teachers edits his/her information which is update by this query.

```

UPDATE ENROLLS x
SET x.STATUS='Pending Delete'
WHERE x.STUDENT_ID='${id}' AND x.SESSION_NAME='${session_name}' AND x.COURSE_ID='${course_id}'

```

Students sends his request to drop a course to admin for approval.

## 3. Delete

```

if (action === "DROP") {
  if (status === "Pending") {
    await run(
      DELETE FROM ENROLLS WHERE STUDENT_ID = '${id}' AND COURSE_ID = '${course_id}' AND SESSION_NAME='${session_name}'
    );
  }
}

```

Admin approves to drop a course for a student which have status Like 'pending'

```
DELETE FROM TEACHER
WHERE TEACHER_ID like '${idToDelete}';
END;
```

Delete a teacher (it also removes the teacher from the Teaches table);

```
await run(`DELETE FROM Enrolls
WHERE Student_ID LIKE '${studentID}' AND COURSE_ID LIKE  '${course_ID}' AND Status LIKE 'Pending'`);
```

Delete student from Enrolls table directly if his course status is 'pending'

#### 4. Advanced Query :

```
`SELECT C.Course_ID, C.Course_Title, E.Session_Name, C.Level_Term, C.Credit_Hours, R.Grade
FROM Enrolls E
INNER JOIN (
    SELECT Course_ID, Grade
    FROM Results
    WHERE Student_ID LIKE '${id}'
) R ON E.Course_ID = R.Course_ID
INNER JOIN Courses C ON E.Course_ID = C.Course_ID
WHERE E.Student_ID = '${id}'
AND C.Level_Term = '${strActual}'`
```

Gives grades of a particular terms courses(3 tables join)

```
(`SELECT C.Course_Title, C.Course_ID, C.Level_Term,S.SESSION_NAME ,C.CREDIT_HOURS
FROM Courses C
INNER JOIN Sessions S ON C.Course_ID = S.Course_ID
WHERE C.Department Like '${dept}' AND S.Session_Name like '${sessionName}'
MINUS
(SELECT C.Course_Title,C.Course_ID, C.Level_Term, T.Session_Name,C.CREDIT_HOURS
FROM Teaches T
INNER JOIN Courses C ON T.Course_ID = C.Course_ID
WHERE T.Teacher_ID LIKE '${id}' AND T.Session_Name like '${sessionName}' AND
(T.Status LIKE 'Current' OR T.Status LIKE 'Pending' OR T.Status LIKE 'Pending Delete'))`);
```

To show the available courses of a teacher

```
const students =
    await run(`SELECT DISTINCT S.Student_ID, S.First_Name, S.Last_Name, S.Level_Term,
S.Department, S.Email, S.Phone_No, S.Address, S.Section_name, S.Advisor_ID
FROM Teaches T
INNER JOIN Enrolls E ON T.Course_ID = E.Course_ID AND T.Session_Name = E.Session_Name
INNER JOIN Student S ON E.Student_ID = S.Student_ID
WHERE T.Teacher_ID LIKE '${id}' AND T.Status LIKE 'Current'`);
```

To show the students under a teachers taken courses(3 tables join)

```
let students = await run(`SELECT S.Student_ID
FROM Teaches T
INNER JOIN Enrolls E ON T.Course_ID = E.Course_ID AND T.Session_Name = E.Session_Name
INNER JOIN Student S ON E.Student_ID = S.Student_ID
WHERE T.Teacher_ID LIKE '${id}'
AND T.Course_ID LIKE '${value}'
AND T.Session_Name LIKE '${sessionName}'
AND T.Status LIKE 'Current'`);
```

To show the students under a teacher's particular course

```
const data = await run(`SELECT *
FROM Student s
JOIN Enrolls e ON s.Student_ID = e.Student_ID
WHERE e.Course_ID = '${course_ID}' AND session_name = '${session}'`);
```

To show a student's info along by a course\_Id along with course Id, course title, level\_term, credit\_hours etc