**Data warehouse design Report**

Contents

[**TASK 1: Problems incurred** 2](#_Toc33790827)

[**TASK 2: Granularity** 3](#_Toc33790828)

[**TASK 3: Star Schema** 4](#_Toc33790829)

[**TASK 4: Data Warehouse** 5](#_Toc33790830)

[**SQL Statements for Creating the Dimensions and Fact Tables** 6](#_Toc33790831)

## **TASK 1: Problems incurred**

The issue noticed from the data sources are:

1. The STUDENT table appears in two different sources respectively (student information and credit database) with different primary keys. This may suggest that the student table in student information carries different records from the student table in the credit database. Therefore it poses a challenge when creating one dimension for student information as now we have details of students in Silverback indexed by “StudentID” and also we have credit details of students indexed by “StudentNumber” without any foreign key referenced from “studentID” to show that they are the same students.

**Solution**

Create one dimension for student details with primary key as “StudentID” and another dimension name it Previous with primary key “StudentNumber” to store details of students’ previous institution to sever different purposes and also to answer the questions about previous institutions in the requirements of the project.

1. In the credit database there is “Previous\_Studies” table which contains details of students’ previous institution credit points, the problem is that the table’s primary key is same as the student’s table in the credit database (“StudentNumber”), this poses a challenge when determining which table has the most granular data between the two (which one looks up “StudentNumber” from the other or which one will be used as the dimension to lookup “StudentNumber” for the credit points in the Enrolment fact table).

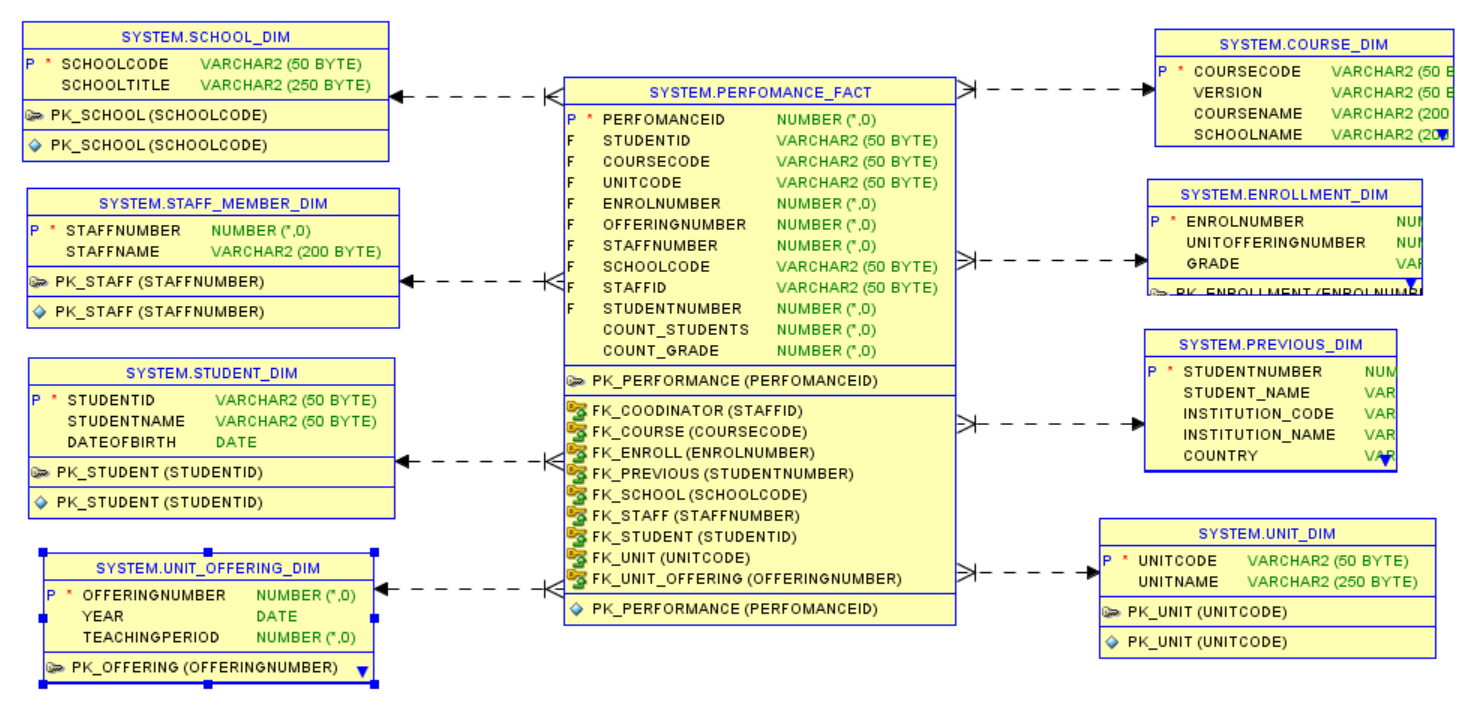
**Solution**

My solution suggestion involves creating one dimension that holds students previous details and has the “StudentNumber” as the primary key. This gives data about the same students but different occasions which also will provide a lookup for students’ previous credit details, instead of creating two dimensions.

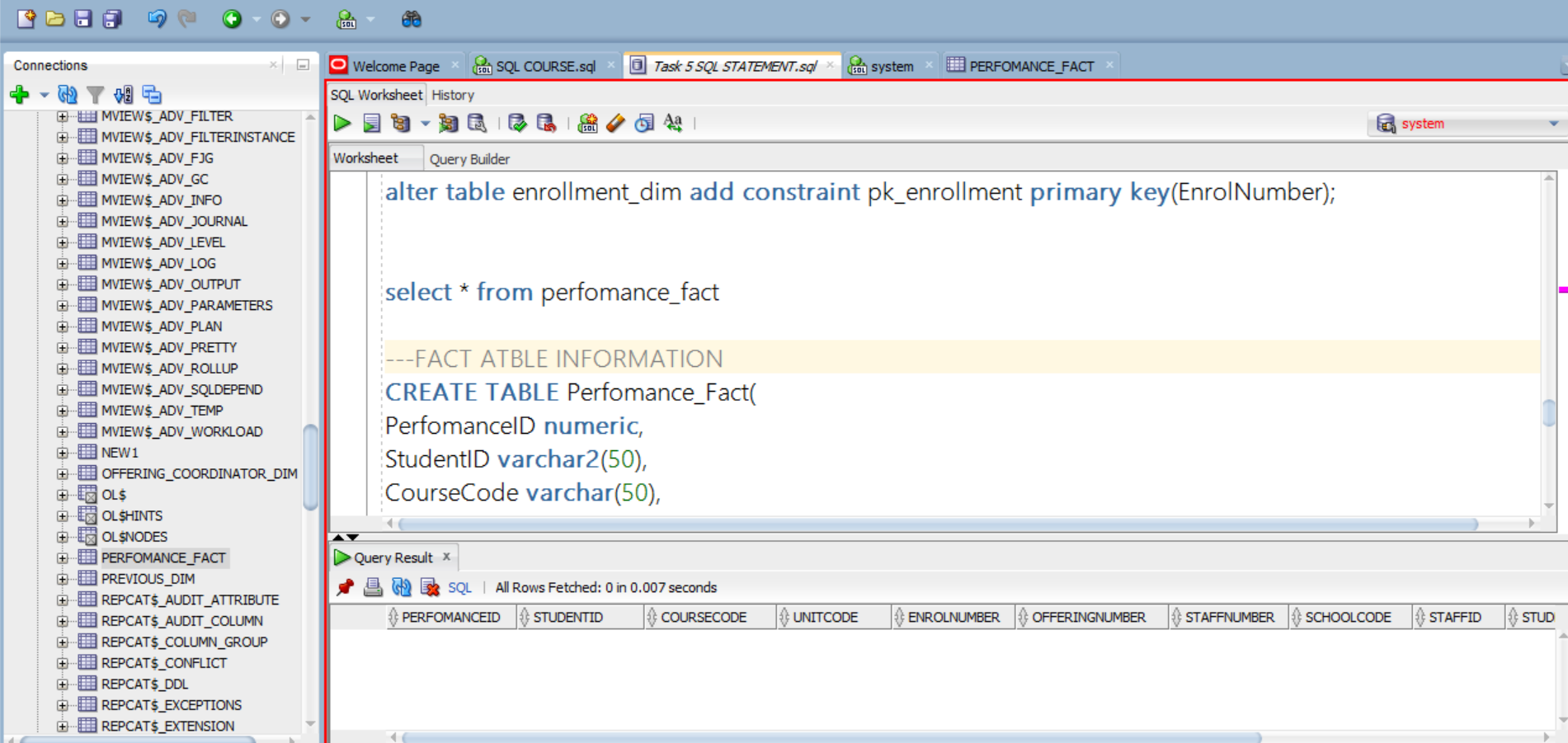
## **TASK 2: Granularity**

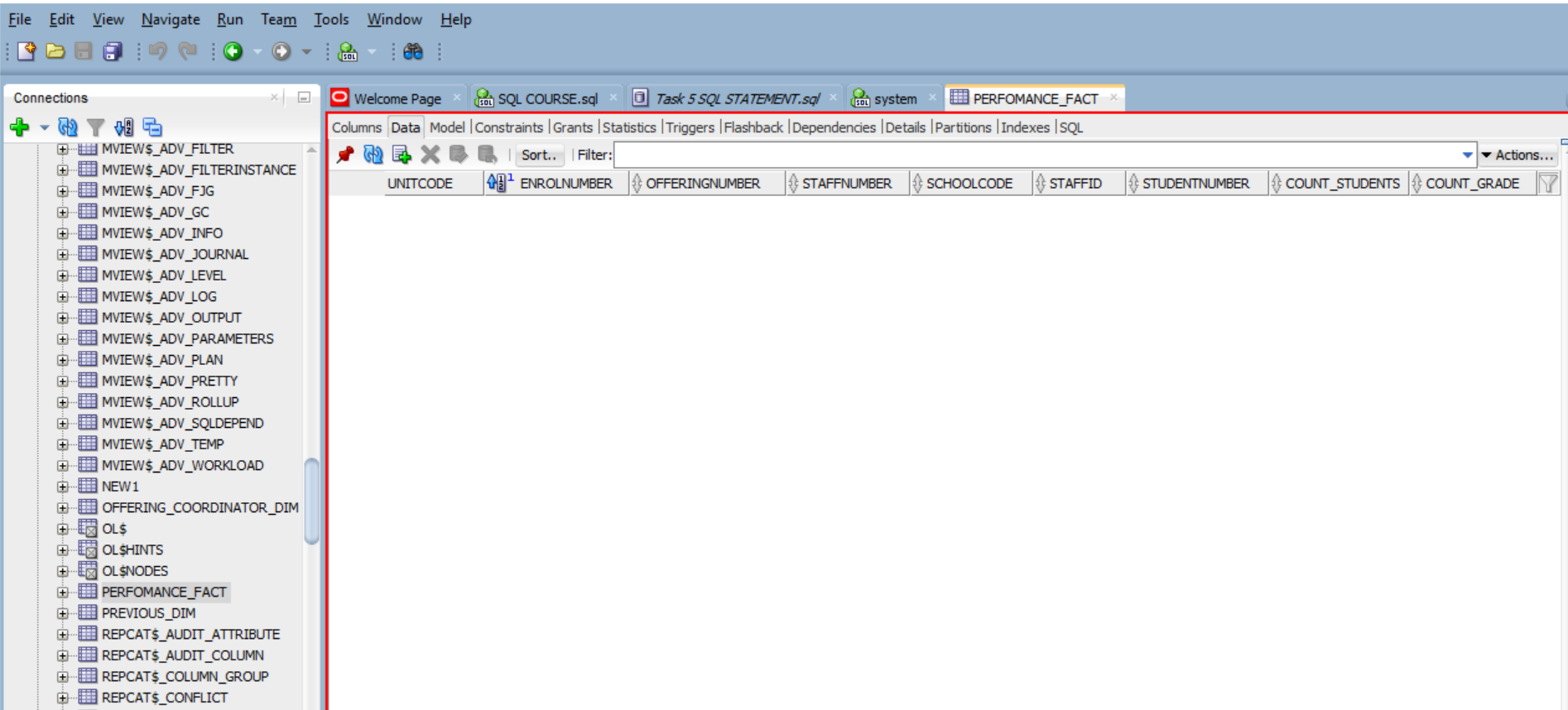
For the data warehouse solution objectives, the number of students per unit, course, per year, per institution and per unit offering. This indicates that we need to store the students’ data in unit granularity in the fact table. In order to generate reports on a unit level the data has to be stored at unit level granularity. If the facts are stored at higher granularity such as course, school, institution or country then to access number of students per unit will be challenging. So in our design we decided to keep records at unit level and year level as per the questions asked.

## **TASK 3: Star Schema**



## **TASK 4: Data Warehouse**





### **SQL Statements for Creating the Dimensions and Fact Tables**

--student information

create table STUDENT\_Dim(

StudentID varchar2(50),

StudentName varchar2(50),

DateOfBirth date

);

alter table student\_dim add constraint pk\_student primary key(StudentID);

---course information

CREATE TABLE COURSE\_Dim(

CourseCode varchar(50),

Version varchar(50),

CourseName varchar(200),

SchoolName varchar(200)

);

alter table course\_dim add constraint pk\_course primary key(CourseCode);

--unit information

CREATE TABLE UNIT\_Dim(

UnitCode varchar(50),

UnitName varchar(250)

);

alter table unit\_dim add constraint pk\_unit primary key(unitcode);

CREATE TABLE UNIT\_OFFERING\_Dim (

OfferingNumber numeric,

Year date,

TeachingPeriod numeric

);

alter table UNIT\_OFFERING\_Dim add constraint pk\_offering primary key(OfferingNumber)

CREATE TABLE OFFERING\_COORDINATOR\_Dim(

StaffID varchar(50),

UnitOfferingNumber numeric

);

alter table OFFERING\_COORDINATOR\_Dim add constraint pk\_coordinator primary key(StaffID)

--HR information

CREATE TABLE STAFF\_MEMBER\_Dim(

StaffNumber numeric,

StaffName varchar(200)

);

alter table STAFF\_MEMBER\_Dim add constraint pk\_staff primary key(StaffNumber)

CREATE TABLE SCHOOL\_Dim(

SchoolCode varchar(50),

SchoolTitle varchar(250)

);

alter table SCHOOL\_Dim add constraint pk\_school primary key(SchoolCode)

--previous institution information

CREATE TABLE PREVIOUS\_Dim (

StudentNumber numeric,

Student\_Name varchar(200),

Institution\_Code varchar(50),

Institution\_Name varchar(50),

Country varchar(50)

);

alter table PREVIOUS\_Dim add constraint pk\_previous primary key(StudentNumber)

CREATE TABLE ENROLLMENT\_Dim(

EnrolNumber NUMERIC ,

UnitOfferingNumber numeric,

Grade varchar(50)

);

alter table enrollment\_dim add constraint pk\_enrollment primary key(EnrolNumber);

---FACT ATBLE INFORMATION

CREATE TABLE Perfomance\_Fact(

PerfomanceID numeric,

StudentID varchar2(50),

CourseCode varchar(50),

UnitCode varchar(50),

EnrolNumber NUMERIC ,

OfferingNumber numeric,

StaffNumber numeric,

SchoolCode varchar(50),

StaffID varchar(50),

StudentNumber numeric,

Count\_students numeric,

Count\_grade numeric,

constraint pk\_performance primary key(PerfomanceID),

constraint fk\_student foreign key(StudentID) references STUDENT\_Dim(StudentID),

constraint fk\_course foreign key(CourseCode) references COURSE\_Dim(CourseCode),

constraint fk\_unit foreign key(UnitCode) references UNIT\_Dim(UnitCode),

constraint fk\_enroll foreign key(EnrolNumber) references ENROLLMENT\_Dim(EnrolNumber),

constraint fk\_unit\_offering foreign key(OfferingNumber) references UNIT\_OFFERING\_Dim(OfferingNumber),

constraint fk\_coodinator foreign key(StaffID) references OFFERING\_COORDINATOR\_Dim(StaffID),

constraint fk\_staff foreign key(StaffNumber) references STAFF\_MEMBER\_Dim(StaffNumber),

constraint fk\_school foreign key(SchoolCode) references SCHOOL\_Dim(SchoolCode),

constraint fk\_previous foreign key (StudentNumber) references PREVIOUS\_Dim(StudentNumber)

);