

CMPG 311 Project 2022

Submission date: 27/05/2022

**Table of Contents**

[**Project Phase 1 – Database Initial Study**](#_35vliorvnk7q) **4**

[1. Members of the group](#_r5lr3b8qvh86) 4

[2. Analyze company situation](#_f3u99lagnt99) 6

[Company Operations:](#_ds6iwkmecdfx) 6

[Company Objectives:](#_56kvdh6z9a5e) 6

[Company Organizational Structure:](#_bu54qdggbs2w) 7

[3. Define problems and constraints](#_fl15o0r25130) 8

[Problems:](#_di3n02w3lqbu) 8

[Constraints:](#_9eoosi60v025) 8

[4. Database system specification](#_9kwsemeesd3i) 10

[Defining Objectives:](#_rs8ysxy8et4m) 10

[Information:](#_402ms9tqz6i8) 10

[Scope:](#_n4hkjva3ys9y) 11

[Boundaries:](#_e0v4pgfnejup) 11

[**Project Phase 2 – Database Design**](#_l4ipa1ljnyj) **13**

[1. Conceptual design](#_y10cm8ql5o8v) 13

[1.1 Business rules](#_174jg1524ocp) 13

[1.2 ER diagram](#_ipc9c0ij4zjz) 14

[2. Logical design](#_yvt7f9if89x4) 15

[2.1 Logical Data Model](#_1y03elnoxho5) 15

[**Project Phase 3 – Physical Design**](#_ehnchuyvo127) **17**

[1. Database objects](#_6m863btpemzk) 17

[1.1 Tables](#_j34btrj47nza) 17

[DROP TABLES](#_dhebvbp8gml0) 17

[DROP SEQUENCE](#_j8u1b3lta9sa) 17

[CREATE TABLE EMPLOYEE](#_qk4jy6it6z4h) 18

[CREATE SEQUENCE](#_2u6urt9diiyh) 18

[CREATE TABLE ADMINISTRATOR](#_zd3pssdkmsba) 19

[CREATE TABLE TEACHER](#_y3cf5eqhg2mb) 19

[CREATE SEQUENCE](#_ukd812u4xwu2) 20

[CREATE TABLE APPLICATION](#_7uw6t4dygv47) 20

[CREATE SEQUENCE](#_epx41zlfh3oy) 21

[CREATE TABLE SCHOOL\_FEES](#_iuq1vwv07vfp) 21

[CREATE SEQUENCE](#_rfjqw21mksnf) 22

[CREATE TABLE DISCIPLINARY\_ACTIONS](#_tzucy2c27sg0) 22

[CREATE SEQUENCE](#_8coi1s1h9kiq) 23

[CREATE TABLE REPORT](#_mmi0g1e6kgc8) 23

[CREATE SEQUENCE](#_xw41gxfowzwd) 24

[CREATE TABLE SUBJECT](#_7gl5to7o3rn5) 24

[CREATE SEQUENCE](#_q9c5eab28sz2) 25

[CREATE TABLE HEADMASTER](#_dcbt3s3ebhk1) 25

[CREATE TABLE TEACHER\_SUBJECT](#_s12f8h78yk81) 26

[CREATE TABLE STUDENT\_SUBJECT](#_wfcj8c4qxopl) 26

[CREATE TABLE REPORT\_SUBJECT](#_pprync2wlizi) 26

[1.2 Indexes](#_54xo4egnaiv3) 27

[1.3 Views](#_v4bvgcqv1gmb) 28

[1.4 Data loading](#_k7bf5gx5qgy) 29

[POPULATE TABLE EMPLOYEE](#_melbeqwcxsx9) 29

[POPULATE TABLE ADMINISTRATOR](#_hntfx2d89rwk) 29

[POPULATE TABLE TEACHER](#_1zl0xantbiww) 30

[POPULATE TABLE STUDENT](#_temdi0je9g5k) 30

[POPULATE TABLE APPLICATION](#_kqce6rhp37se) 31

[POPULATE TABLE SCHOOL\_FEES](#_8k5iuvz3n0zt) 31

[POPULATE TABLE DISCIPLINARY\_ACTIONS](#_a4ugxjgg7pft) 32

[POPULATE TABLE REPORT](#_hg9rfsmsycsa) 32

[POPULATE TABLE SUBJECT](#_gg0n758u13rz) 33

[POPULATE TABLE HEADMASTER](#_qquxxrqhqknh) 34

[POPULATE TABLE STUDENT\_DISCIPLINARY](#_l3q33zc6edy7) 34

[POPULATE TABLE TEACHER\_SUBJECT](#_9cwqrwtjs8rt) 36

[POPULATE TABLE STUDENT\_SUBJECT](#_yf8i9n6jwu4b) 38

[POPULATE TABLE REPORT\_SUBJECT](#_gocea3okxtrc) 39

[2. Queries](#_2rrw0mf1eneh) 40

[2.2 Sorting](#_n3fhywlf4vic) 41

[2.3 LIKE, AND and OR](#_q1or9bxs4hyl) 42

[2.4 Variables and character functions](#_85zghajvrw23) 42

[2.5 Round and/or trunc](#_4ku2grg0x5kr) 43

[2.6 Date functions](#_nj5qny2n87zd) 43

[2.7 Aggregate functions](#_ijtdb1a26g5v) 43

[2.8 Group by and having](#_rqq9nimojyb9) 44

[2.9 Joins](#_idtdxsrwen23) 44

[2.10 Sub-queries](#_3n5x47vu1ytb) 44

# Project Phase 1 – Database Initial Study

## 1. Members of the group

| J.A.B. Louw (Group Leader)  33768080 | D.S. Aucamp  33484112 |
| --- | --- |
| H. van Schalkwyk  31586872 | T. Fourie  34676260 |
| JF. Du Plessis  33642958 | A. Jonker  35023740 |
| L.P. de Beer  32482345 |  |
|  |  |

## 2. Analyze company situation

### Company Operations:

High School Republic is a small sized private school in South-Africa, in the North-West area. The school opened recently with 60 learners and caters learners from grade 8 to 12. The school is generally small in physical size as well as having few staff members under employment. Only 8 teachers, 4 managing figures ; 1 headmaster, 1 Deputy headmaster that handles lower order issues within the school and 2 administrative managers are current members of staff within the school.There are currently 5 classes of 10 students attending the school on any normal day.

The school has shown great potential and thanks to the few learners and teachers, who show great pride in the school, the school is becoming more popular every day.

### Company Objectives:

The school’s initial plan was to accommodate fewer students to ensure every learner gets the necessary attention and education they deserve, the school focuses more on personal growth and how to be prepared for adulthood, while still fulfilling all the criteria a school must adhere to.

Due to the massive request of learners who want to apply, the school decided to expand. When a learner wants to apply to study at the school, a form is filled out including the following; personal information, previous reports, and school related questions. This form is filed and stored for future use. These forms are extracted when school funds need to be paid, reports are written and when disciplinary actions must take place. This also helps to keep track of which students are applying at the school and to see how the students are performing.

Due to the school’s growing popularity, keeping up with the applications and filing is starting to become redundant and impractical. High School Republic would like to implement a database to support the schools day to day operations.

The database would essentially act as a progressive filing system. The goal of the database system is to create ease of access for the application of new students and for members of staff to be able to easily interact and use documents concerning students within the school. The system will create easy management of all information regarding students and their activities.

### Company Organizational Structure:

The headmaster of the school has complete administrative control of the system and has access to edit and view any information on the database. The headmaster also has control of the rights of all members of staff within the database..

The deputy headmaster has complete viewing rights of all information of the database system but cannot edit information without the consent of the headmaster. The deputy reports directly to the headmaster.

The two administrative managers have complete viewing access of the database and all information within it. They have full editing controls as it is their job to organize and manage all information within the schooling environment on the database. Although they have full administrative rights, they can never overrule the instruction of the headmaster and must always report to him. They can be instructed on certain tasks by the deputy.

The eight teachers working within the schooling environment report directly to the deputy headmaster and the headmaster themself. They have viewing access to the database system but can only view certain information, they do not have access to all data. Members of staff have no editing rights within the database system.

## 3. Define problems and constraints

### Problems:

· There is no existing information system to support in the handling of new applications from prospective students.

· The system is still paper based, and the accuracy and validity are not meeting the requirements of the fast-growing school.

· The personnel of the school do not have a way to easily access and handle the student’s personal information, previous reports and payment information, also how each student is progressing.

· The personnel of the school find it really hard to select and accept applying students with their unorganized paper-based system and this wastes a lot of time.

· Mistakes (when using the paper-based system) are made frequently which causes unnecessary confusion and misunderstandings.

· The growing and enormous amount of paper-work the school has to deal with is just unbearable and sometimes leads to errors in accepting students and dealing with other issues, i.e. payments, disciplinary actions etc.

### Constraints:

· The cost of licensing of the program environment and the cost of the new hardware where the program will be implemented.

· The project will have a restricted time period to be implemented as It should be done before the next school year is at hand, which roughly starts around the 10th-15th January.

· Since the school is still using a paper-based procedure to run the system, new hardware must be required to run the information system and with that said, the system will run on Visual Studio and Oracle SQL will be used. Adding to that, Visual Studio licensing will be required for maintenance.

· Teachers who act as internal resources to the project will only be available during daily student lunch breaks.

· A learning curve of the system is also a constraint because not everyone will be able to operate on the system and will need training.

## 

## 4. Database system specification

### Defining Objectives:

The system’s objective is to support the school with daily operations and would act as a progressive filling system. It will improve the quality and management of information of the school and make student’s information more accessible for all faculty members. The school currently works with a paper base system and does not have any other systems in place however the new system will only be able to interface with future systems.

The system will only share data with future systems because it is the first and currently only system that the school has. The headmaster will have complete administrative control and access to any information. The deputy headmaster has complete viewing rights of all information but, cannot edit information without the consent of the headmaster. The two administrative managers have complete viewing access of the database and all information within it. The eight teachers have limited viewing access and no editing rights.

### Information:

Student:

· Application forms

· Registration forms

· Personal information

o Full name and surname

o Birth date and age

o School grade

o Parent/guardian contact information

· Classes

· Grades

Staff:

· Personal information

o Full name and surname

o Birth date and age

o Contact details

· Subjects

· Position

### Scope:

The database system will not cover the entirety of the schools’ business functions but mainly the handling of students’ enrollments, tracking of grades and tracking of payments. With that said the database must contain the information of each individual student registered to the school, this includes the following: Students’ first name, students’ last name, students’ DOB, students’ gender, students’ grade, students’ previous school records, students’ current school records, students’ enrollments. The database needs to be able to add new students that enroll in the school and remove students that have for some reason left the school and be able to edit student details if there were errors for example. The database needs to be able to report students’ current and previous results and must be able to add, remove and manage(edit) marks of the reports. The database should also store payment history of the students and handle new payments.

### Boundaries:

· Finances: Budget of R350000.

· Time: Must be completed before the new school year starts and time must be given for system users to be trained.

· Human Resources: There are 7 team members working on the project so their personal matters and times they are available need to be taken into account to complete the project in time.

· Software/Hardware: Software and hardware need to be not only compatible with the system but must be able to process at acceptable speeds to reduce wasting time.

· Teachers will only be able to view student records and pull relevant reports. Cannot edit.

· The deputy headmaster can view and pull all information of students, but cannot edit.

# 

# Project Phase 2 – Database Design

## 1. Conceptual design

### 1.1 Business rules

* Each student can fill in only one Application form.
* An application form can be filled in by only one student
* Each student can have many subjects.
* Each subject can have many students.
* Each teacher can give many subjects.
* Each subject can be taught by many teachers.
* Each student pays one or many school fees.
* Each school fee is being paid by only one student.
* A student can have zero or many disciplinary actions.
* A disciplinary action can be given to zero or many students.
* Each student can have many reports.
* A report is given to only one student.
* Administrators can administer many students.
* A student can only be administered by 1 administrator.
* A report can have many subjects.
* A subject can be on many reports.

### 1.2 ER diagram

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 2. Logical design

### 2.1 Logical Data Model

**EMPLOYEE (EMPLOYEE\_ID**(PK), EMP\_NAME, EMP\_SURNAME, EMP\_CONTACT\_NUMBER, EMP\_IS\_ADMIN, EMP\_IS\_TEACHER **)**

**ADMINISTRATOR** (***EMPLOYEE\_ID***(PK, FK), ADMIN\_IS\_HEAD)

**TEACHER** (***EMPLOYEE\_ID***(PK, FK))

**HEADMASTER** (***EMPLOYEE\_ID***(PK, FK), HEADMASTER\_TITLE)

**STUDENT (STUDENT\_ID**(PK),*EMPLOYEE\_ID*(FK), STU\_NAME, STU\_SURNAME, STU\_GRADE, STU\_AGE, STU\_CELL)

**SUBJECT** (**SUBJECT\_ID**(PK), SUBJECT\_NAME, SUBJECT\_GRADE, SUBJECT\_PASS, SUBJECT\_SYMBOL)

**TEACHER\_SUBJECT** (***EMPLOYEE\_ID***(PK, FK), ***SUBJECT\_ID***(PK, FK))

**STUDENT\_SUBJECT** (***STUDENT\_ID***(PK, FK), ***SUBJECT\_ID***(PK, FK))

**SCHOOL\_FEES** (**FEES\_ID**(PK), *STUDENT\_ID*(FK), FEE\_DESC, FEE\_AMOUNT, FEE\_PAID, FEE\_DATE)

**STUDENT\_DISCIPLINARY** (***STUDENT\_ID***(PK, FK), ***DISCIPLINE\_ID***(PK, FK))

**DISCIPLINARY\_ACTIONS (*DISCIPLINE\_ID***(PK), *STUDENT\_ID*( FK), DISC\_DESCRIPTION, DISC\_ACTION\_TO\_TAKE)

**REPORT** (**REPORT\_ID**(PK), *STUDENT\_ID*(FK), TERM\_NUMBER, TERM\_PASS)

**REPORT\_SUBJECT** (***REPORT\_ID***(PK, FK), ***SUBJECT\_ID***(PK, FK))

**APPLICATION** (**APPLICATION\_ID**(PK), *STUDENT\_ID*(FK),APPLICATION\_DATE, APPLICATION\_APPROVED)

# 

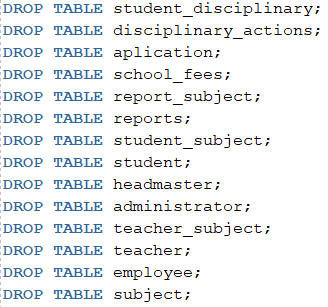
# Project Phase 3 – Physical Design

## 1. Database objects

### 1.1 Tables

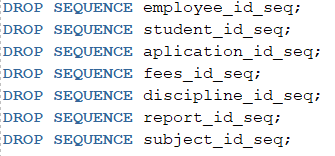
#### DROP TABLES

The DROP TABLE clause was used to drop all the tables we have created, therefore removing them ensures that there will be no duplicate tables. All 14 of our tables are dropped and the clauses are arranged in such a way that the child entities are first removed before the parent entities to ensure integrity.



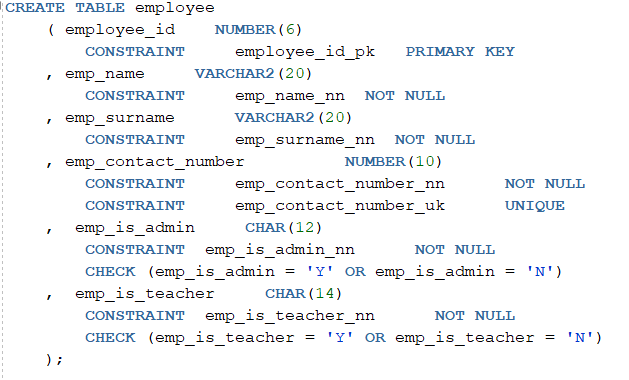
#### DROP SEQUENCE

The DROP SEQUENCE clause was used to remove any current sequence on the primary keys to ensure that when the database is populated the sequences all start where they are supposed to. Without the DROP SEQUENCE clause the identifiers will continue to increment when the script is run.



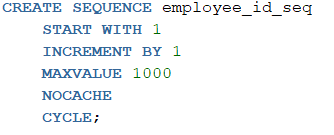
#### CREATE TABLE EMPLOYEE

Employee is a parent entity with child entities Administrator and Teacher. For the primary key (employee\_id) we used a number data type with space for 6 integers to distinguish between the employees. All columns that contain names and surnames make use of the VARCHAR2 data type and where only one character needs to be entered a CHAR data type is used. A CHECK statement is used to ensure the user entered either a “Y” or a “N” character when stating if the employee is an admin or a teacher. NOT NULL constraints were used to ensure that the user in fact enters a value where really needed. The UNIQUE clause was also used to ensure that the employees' contact numbers are not the same.

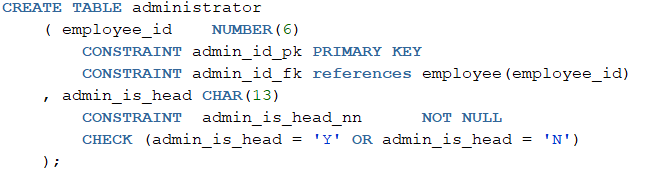


##### CREATE SEQUENCE

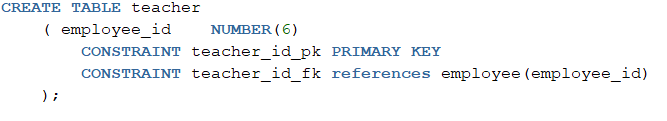
Created a sequence for the primary key of the Employee table (employee\_id) to ensure that the first employee that is inserted into the database starts at one and the ID value is incremented by one with every new employee added. Once it reaches the max value it begins the cycle from the start.



#### CREATE TABLE ADMINISTRATOR

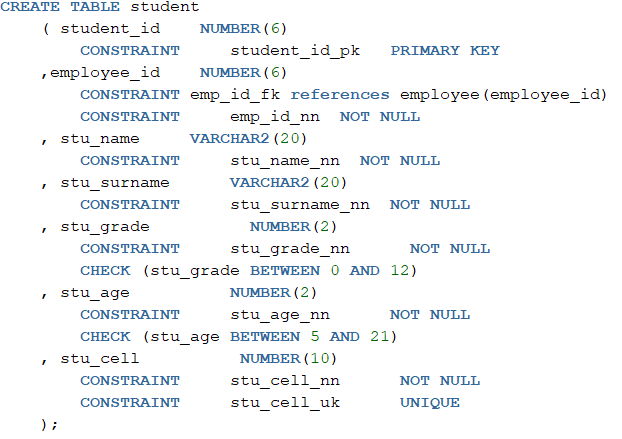
Administrator is one of the child entities of the Employee table. It consists of a Foreign Primary Key (employee\_id) that uses a number data type with space for 6 integers to distinguish between the administrators. Admin\_is\_head is a CHAR datatype used to check if the current admin is the headmaster. NOT NULL clauses were used to ensure the users entered values where needed.

#### CREATE TABLE TEACHER

Teacher is one of the child entities of the Employee table. It consists of a Foreign Primary Key (employee\_id) that uses a number data type with space for 6 integers to distinguish between the teachers.

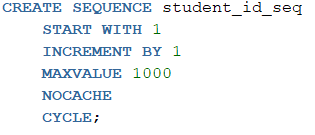
CREATE TABLE STUDENT

The student table contains all the personnel details of the student. For the primary key (student\_id) we used a number data type with space for 6 integers to distinguish between the students. All columns that contain names and surnames make use of the VARCHAR2 data type. A CHECK statement is used to ensure the user entered a number between the specified values when stating the students age and grade. NOT NULL constraints were used to ensure that the user in fact enters a value where really needed. The UNIQUE clause was also used to ensure that the students' contact numbers are not the same.



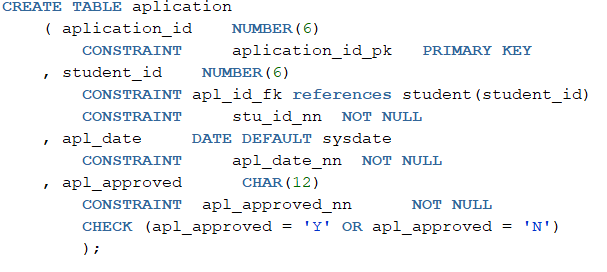
##### CREATE SEQUENCE

Created a sequence for the primary key of the Student table (student\_id) to ensure that the first student that is inserted into the database starts at one and the ID value is incremented by one with every new student that is added. Once it reaches the max value it begins the cycle from the start.



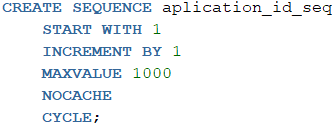
#### CREATE TABLE APPLICATION

The application table contains information on whether or not the students’ application was accepted and the date the application was issued. For the primary key (aplication\_id) we used a number data type with space for 6 integers to distinguish between the applications. Student\_id is used as a foriegn key to establish the relation between application and student. The date of the application is stored using the DATE clause and the system’s date.



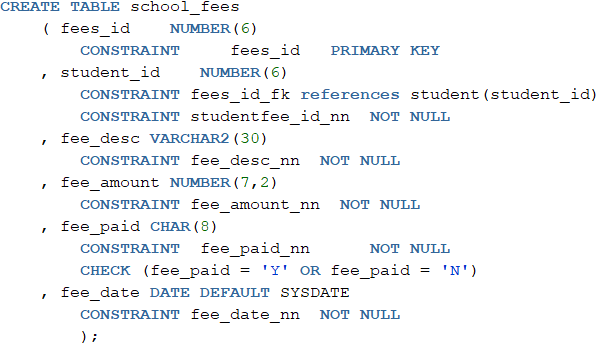
##### CREATE SEQUENCE

Created a sequence for the primary key of the Application table (aplication\_id) to ensure that the first application that is inserted into the database starts at one and the ID value is incremented by one with every new application that is added. Once it reaches the max value it begins the cycle from the start.



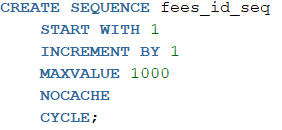
#### CREATE TABLE SCHOOL\_FEES

The school fees table contains information on the fees the students have paid or still have to pay. For the primary key (fees\_id) we used a number data type with space for 6 integers to distinguish between the fees. Student\_id is used as a foriegn key to establish the relation between school fees and students. NOT NULL constraints were used to ensure that the user in fact enters a value where really needed. Where only one character needs to be entered a CHAR data type is used. A CHECK statement is used to ensure the user entered either a “Y” or a “N” character when stating if the student paid. For the money variables the NUMBER data type was used and stated to round to 2 decimal figures.



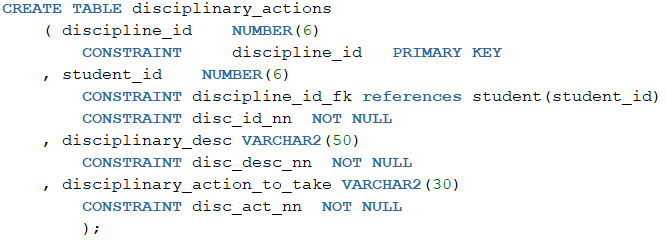
##### CREATE SEQUENCE

Created a sequence for the primary key of the school fees table (fees\_id) to ensure that the first fee that is inserted into the database starts at one and the ID value is incremented by one with every new fee that is added. Once it reaches the max value it begins the cycle from the start.



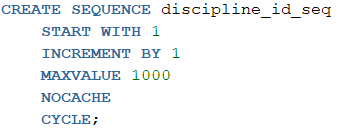
#### CREATE TABLE DISCIPLINARY\_ACTIONS

The disciplinary actions table contains information on which students have received disciplinary actions and for what reason. For the primary key (discipline\_id) we used a number data type with space for 6 integers to distinguish between the disciplines. Student\_id is used as a foriegn key to establish the relation between disciplinary actions and students. VARCHAR2 is used to describe the fault of the student and the disciplinary actions taken.



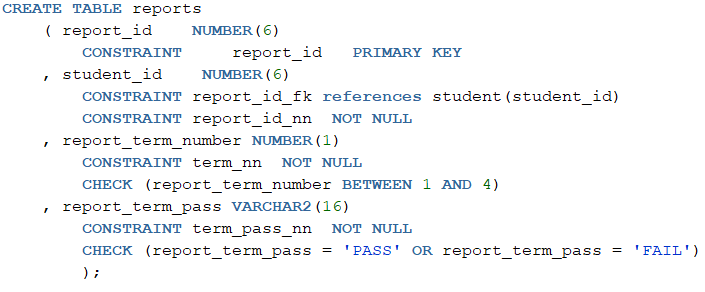
##### CREATE SEQUENCE

Created a sequence for the primary key of the disciplinary actions table (discipline\_id) to ensure that the first disciplinary action that is inserted into the database starts at one and the ID value is incremented by one with every new disciplinary action that is added. Once it reaches the max value it begins the cycle from the start.



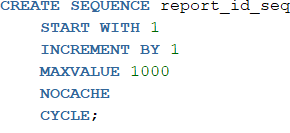
#### CREATE TABLE REPORT

The report table contains the current term of the school year as well as whether or not the student has passed the term. For the primary key (report\_id) we used a number data type with space for 6 integers to distinguish between the reports. Student\_id is used as a foriegn key to establish the relation between reports and students. To indicate the term a NUMBER datatype with the space for one integer is used along with the CHECK clause to ensure the user enters a value between 1 and 4. The VARCHAR2 datatype was used along with a CHECK clause to ensure the user enters “PASS” or “FAIL”.



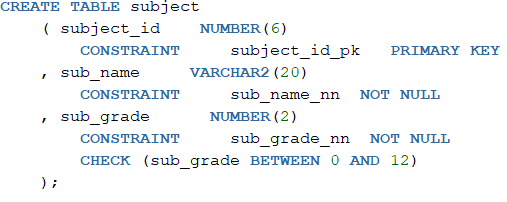
##### CREATE SEQUENCE

Created a sequence for the primary key of the reports table (report\_id) to ensure that the first report that is inserted into the database starts at one and the ID value is incremented by one with every new report that is added. Once it reaches the max value it begins the cycle from the start.



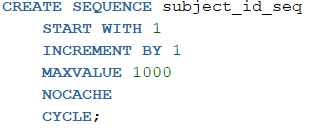
#### CREATE TABLE SUBJECT

The subject table contains all the subjects the school has to offer. For the primary key (subject\_id) we used a number data type with space for 6 integers to distinguish between the subjects. VARCHAR2 was used to state the subject name and for sub\_grade a NUMBER datatype along with a CHECK clause is used to ensure the correct subject grade was entered.



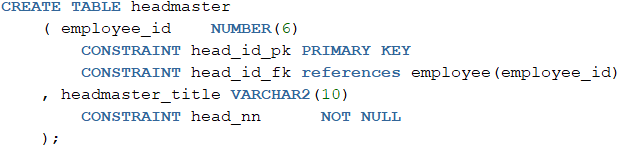
##### CREATE SEQUENCE

Created a sequence for the primary key of the subject table (subject\_id) to ensure that the first subject that is inserted into the database starts at one and the ID value is incremented by one with every new subject that is added. Once it reaches the max value it begins the cycle from the start.



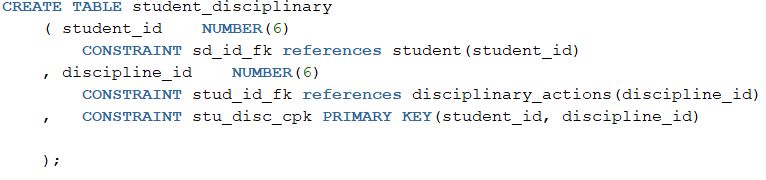
#### CREATE TABLE HEADMASTER

The table headmaster contains the title of the headmaster using the VARCHAR2 datatype and a NOT NULL constraint. Employee\_id is a Foreign Primary Key used to establish the relation between employee and headmaster.



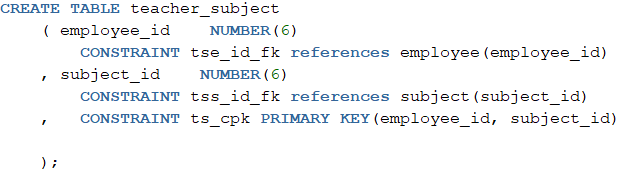
CREATE TABLE STUDENT\_DISCIPLINARY

The student\_disciplinary table is a bridge entity used to break the many-to-many relationship found between the student and the disciplinary actions tables. It contains two Foreign Primary Keys (student\_id, discipline\_id) that establishes the relations between the student table and the disciplinary actions table.



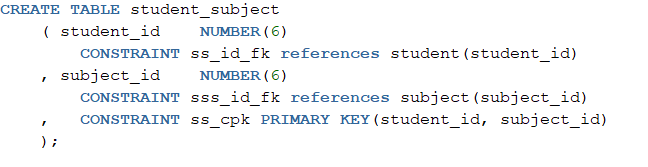
#### CREATE TABLE TEACHER\_SUBJECT

The teacher\_subject table is a bridge entity used to break the many-to-many relationship found between the teacher and the subject table. It contains two Foreign Primary Keys (employee\_id, subject\_id) that establish the relations between the subject table and the teacher table.



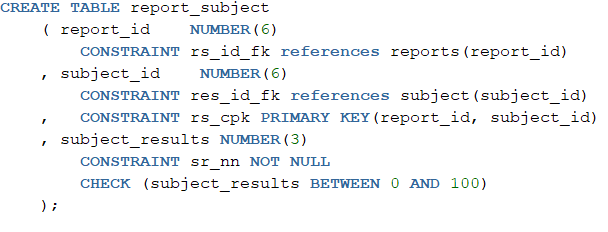
#### CREATE TABLE STUDENT\_SUBJECT

The student\_subject table is a bridge entity used to break the many-to-many relationship found between the student and the subject table. It contains two Foreign Primary Keys (student\_id, subject\_id) that establish the relations between the student table and the subject table.



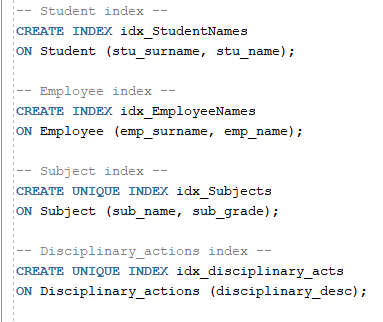
#### CREATE TABLE REPORT\_SUBJECT

The report\_subject table is a bridge entity used to break the many-to-many relationship found between the report and the subject table. It contains two Foreign Primary Keys (report\_id, subject\_id) that establish the relations between the report table and the subject table. This table also contains the result of a certain subject for a certain student. The results are stored as NUMBER data types allowing for 3 integers. A CHECK clause is used to ensure the user enters a value between 0 and 100 for the results.



### 1.2 Indexes

We created indexes on the Student, Employee, Subject and Disciplinary\_actions tables. The goal of this is to increase the speed at which we can retrieve certain information from database tables. For example, on the student table an index is created on the student’s surname and name to quickly access information about a specific student and the same with the other tables.



### 

### 1.3 Views

A view is a technique used to speed up filtering through the database tables. All of the views have an obvious goal to them, for example, the “Students\_failing” view is used to quickly get a view of all of the students that are currently failing their terms. The “Admin\_Employees” view displays all of the employees that are system-admins, the “Students\_disciplinary\_actions” view is used to retrieve all students that have a disciplinary action against them.



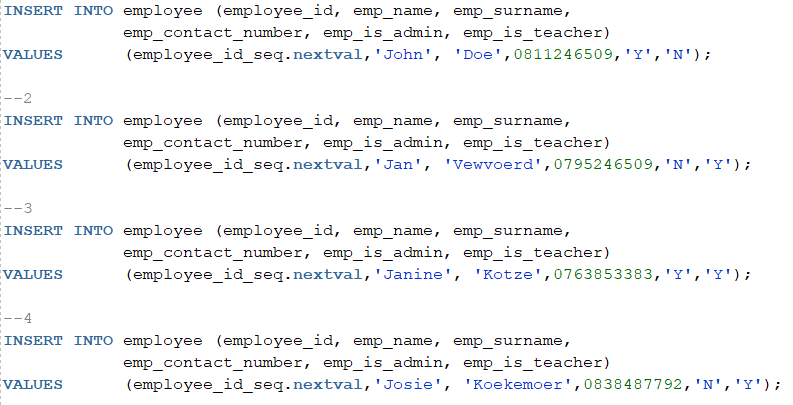
### 

### 

### 1.4 Data loading

* After creating the tables we loaded the data into the tables using simple INSERT INTO statements.
* All datatypes of the fields inserted correspond with the datatypes used in the creation of the tables. Statements were initialized in the table creation to check if the data that is inserted is correct.
* An automatic numbering statement (seq.nextval) is used to make sure that each primary key is unique.
* The keywords ‘Y’ and ‘N’ are used to indicate a YES and NO.
* When a date is required the system date is automatically inserted using the (DATE and SYSDATE) statement.

#### POPULATE TABLE EMPLOYEE



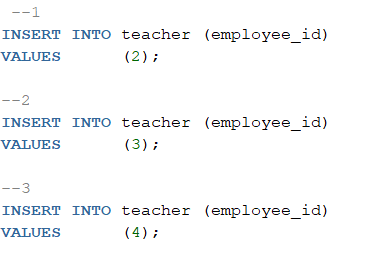
#### POPULATE TABLE ADMINISTRATOR

#### 

#### 

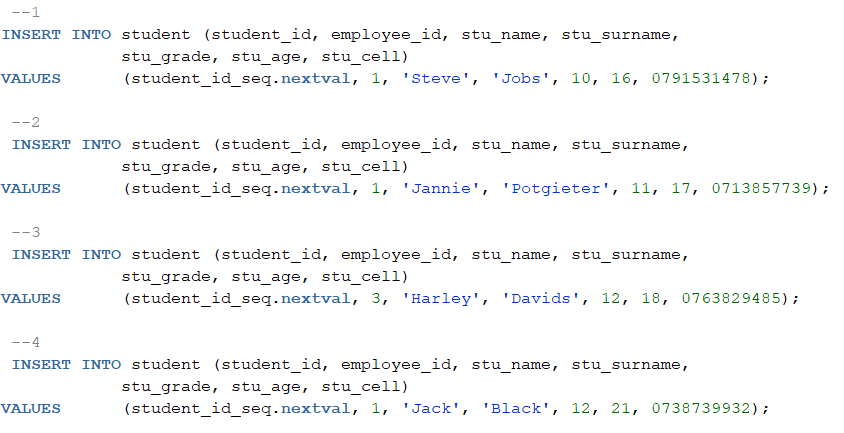
#### POPULATE TABLE TEACHER

#### 



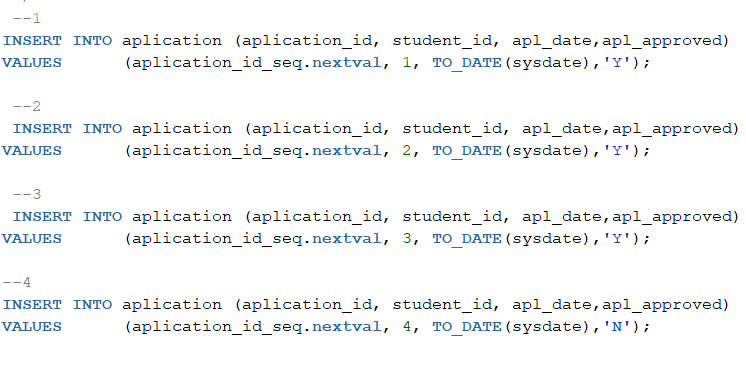
#### POPULATE TABLE STUDENT

#### 



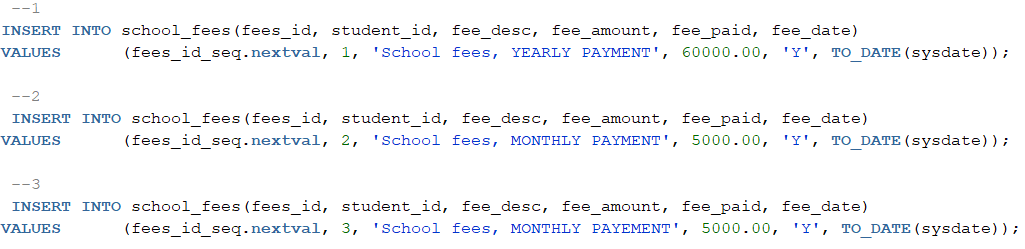
#### POPULATE TABLE APPLICATION

#### 



#### POPULATE TABLE SCHOOL\_FEES

#### 



#### POPULATE TABLE DISCIPLINARY\_ACTIONS

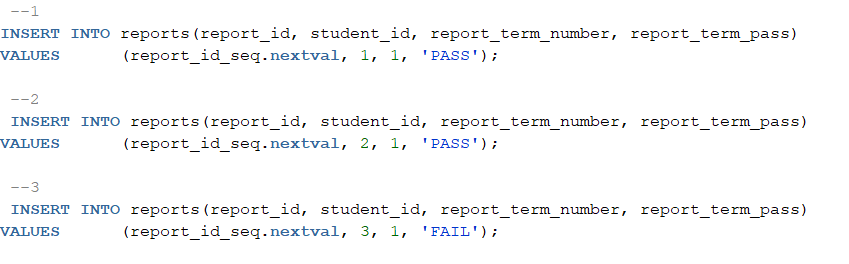
#### 

#### 

#### 

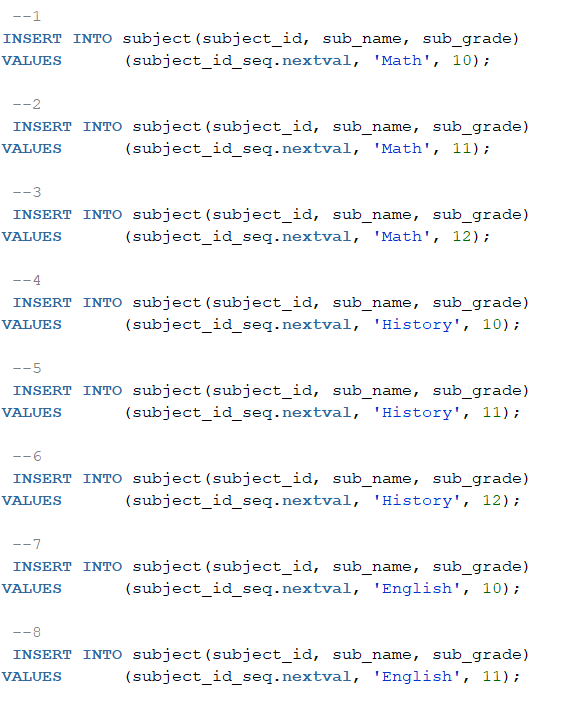
#### POPULATE TABLE REPORT

#### 



#### POPULATE TABLE SUBJECT

#### 

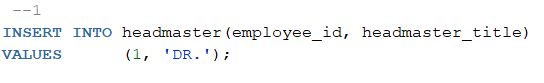




#### 

#### POPULATE TABLE HEADMASTER

#### 



#### 

#### 

#### 

#### 

#### 

#### 

#### 

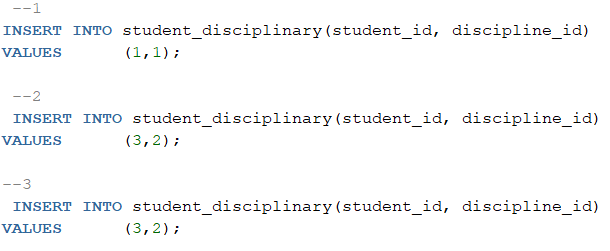
#### 

#### 

#### 

#### 

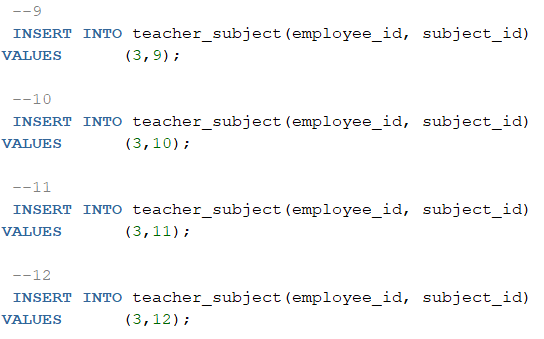
#### POPULATE TABLE STUDENT\_DISCIPLINARY



POPULATE TABLE TEACHER\_SUBJECT

#### 





#### POPULATE TABLE STUDENT\_SUBJECT

#### 



#### POPULATE TABLE REPORT\_SUBJECT



## 2. Queries

Perform queries on the data loaded into the database to illustrate your knowledge and skills on each of the requirements below (you may use your imagination to think of an appropriate scenario where you will perform such a query). Provide the scenario or a description of the purpose of each query along with the query and the output thereof.

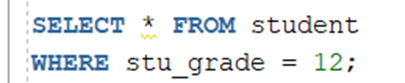
2.1 Limitation of rows and columns

The following statements are needed by the company:

* This statement shows all students enrolled in the school.



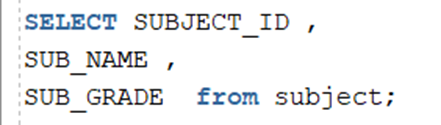
* This statement gives all students that are currently in Grade 12 at the school.



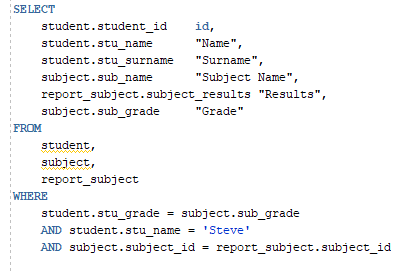
* This statement shows all employees currently working at the school.



* This statement shows all the subjects that the school offers and the grades that it is available to as a subject.



* This statement gives the information on a specific student (aka Steve).

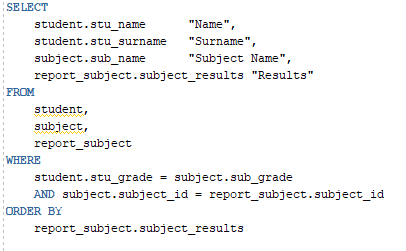


* This statement shows all information related to school fees that only the administrator may view.

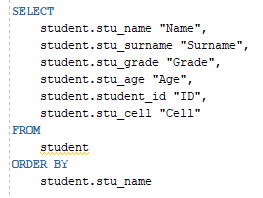


### 2.2 Sorting

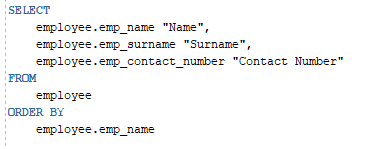
* This statement gives a sorted table containing the subject result.



* This statement sorts all the students by their name.

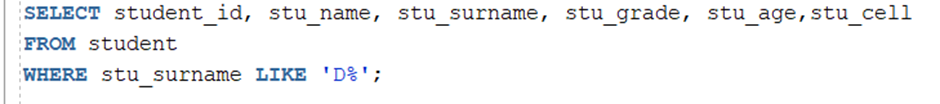


* This statement sorts all the employees by their name.

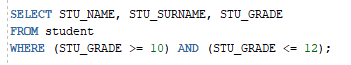


### 2.3 LIKE, AND and OR

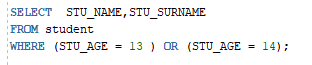
* This query can be used to look for any student and all their information if the student’s information is needed. Any employee is privileged to see this information and thus no restrictions are needed. The information can be searched for with only the first letter of their surname. Currently a teacher was looking for all students that have a surname starting with “D”.



* This statement shows all students that are between grade 10 and 12.

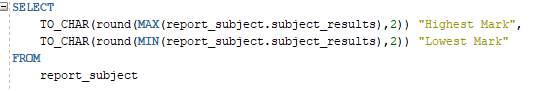


* This statement shows all students that could possibly be in grade 8 even if they are sent to school early or late.



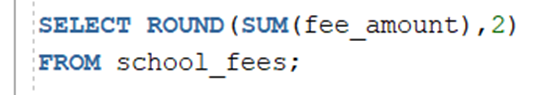
### 2.4 Variables and character functions

* This statement shows the highest and lowest mark.



### 2.5 Round and/or trunc

* This statement allows for all fees received for the entirety of the year so far, for all students that have paid their school fees.



### 2.6 Date functions

* This statement shows all applications created in the last 30 days.

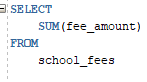


### 2.7 Aggregate functions

* This statement returns the total number of employees.



* This statement shows the sum of the school fees.



* This statement shows the average school fees of students.

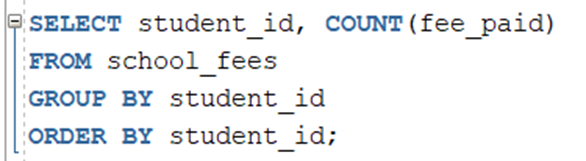
### 

* These two statements return the lowest and highest school fees out of all students.

### 

### 2.8 Group by and having

* This statement is used to look for any outstanding school fees for all students. By using the Count function on fee\_paid it can show us the amount of times.



### 

### 2.9 Joins

* The statement shows all employees that are administrators.



### 2.10 Sub-queries

* This statement is used to retrieve all the students that are in the school younger than the required age of grade 12, which is 18. It also gives an indication of how many students will be leaving the school at the end of the year and gives an indication of new students that can be enrolled in the school the following year.



* The queries for the Views created.

.

