Project work I: Oracle

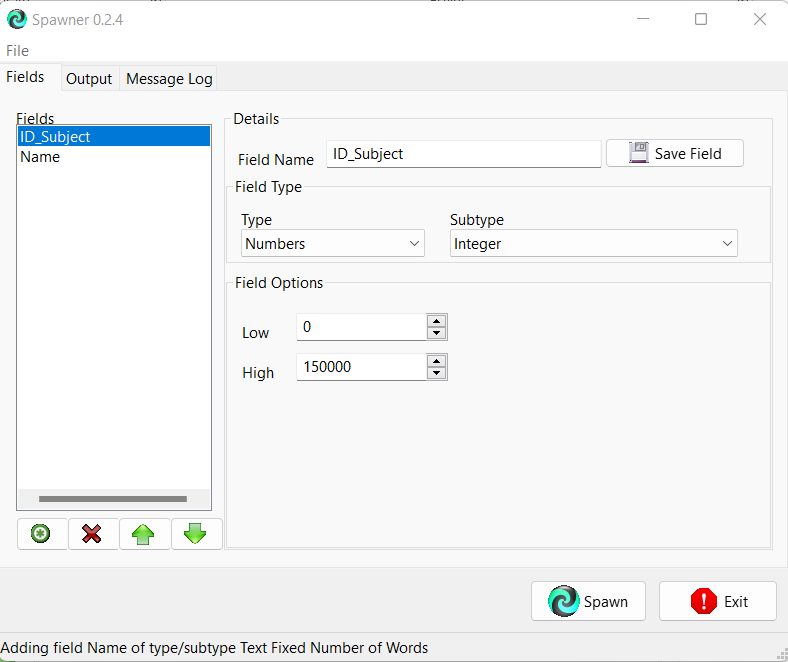
**Name:** Zinga Banda Firmino René **Neptun:** GAA4QU

1. **Generate or Create Database**

I created the 6 tables which are the elements of my database using Spawner application, I create 6 tables which is **Student**, **Student\_Subject**, **Subject**, **Student\_SW**, **Scientific\_work** and **Award** and publisher with **0.1 million records on Student** table**, 0.2 million records in Student\_Subject** table **and 0.1 million records on Student\_Subject.**

First, open Spawner and I made two field, ID\_Subject and Name on the table Subject, After fill the field name, I choose type of the field, this type depends on the field that I will made. So, for Subject table, I set my field like this:

* Field Name: ID\_Subject, Type: Numbers, Subtype: Integer, Low Number: 0, High Number: 150000 and another field is,
* Field Name: Name, Type: Text, Subtype: Fixed Number of Words, Fields option: 5 words.



Uma imagem com texto, captura de ecrã, ecrã, software

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, software, ecrã

Descrição gerada automaticamente

After finish with the field, go to Output tab, set the number of records to generate and set the name output file. In this example, I made 0.15 million records for Subject table, and Subject.csv for the name of output file. And click Include field ID\_Subject as first row, I use this so my first row in table will be the ID\_Subject of the field.

As you can see above, Spawner generated 0.15 million records data for me in csv file format. And when I open the table in excel, it will look like this:

Uma imagem com texto, captura de ecrã, número, file

Descrição gerada automaticamente

I repeat this process generate database objects for other 5 tables:

* 0.2 million records for **Student\_Subject** table with filed name: **Cod\_Student**, **ID\_Subject** and **Grade**.
* 0.1 million records for **Student** table with field name: **Cod\_Student**, **Name**, **Age**, **Genre**, **Group.**
* 1500 records for **Student\_Work** table with field name: **Cod\_Student**, **ID\_SW**, **Committee**.
* 200 records for **Scientific\_Work** table with field name: **ID\_SW**, **Title**, **Category**, **Page**.
* 1000 records for **Award** table with field name: **ID\_A**, **Category**, **ID\_SW**.

1. **Get and Load Database**
   1. **Get/Create Table**

After I have my table database, open SQL Developer, connect to database from ELTE, in this project I use ***ulmatunnel*** connection. After I connect to the database, I can do load my data to SQL Developer. First, create new table using shell script and since I have 6 tables, I used 6 script to make 6 tables.

|  |
| --- |
| CREATE TABLE gaa4qu.Subject (  IDS INTEGER NOT NULL,  name VARCHAR2(50 CHAR)  )  PCTFREE 10 PCTUSED 40 TABLESPACE users LOGGING  STORAGE ( INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS 2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT )  NO INMEMORY;  ALTER TABLE gaa4qu.Subject ADD CONSTRAINT Subject\_pk PRIMARY KEY ( IDS );  CREATE SEQUENCE GAA4QU.Subject\_id\_SEQ  START WITH 1  NOCACHE  ORDER ;  CREATE OR REPLACE TRIGGER GAA4QU.Subject\_id\_TRG  BEFORE INSERT ON GAA4QU.Subject  FOR EACH ROW  WHEN (NEW.ids IS NULL)  BEGIN  :new.ids := gaa4qu.Subject\_id\_seq.nextval;  end;  / |

*Script for Subject table*

*CREATE TABLE gaa4qu.Subject* is using for create new table with name Subject in gaa4qu account which is my account with field id data type integer and set no null for primary key; name data type varchar set in size about 50 character.

* *CREATE INDEX* gaa4qu*.Subject\_\_idx ON* is using for creating index for Subject table using field name so it can be use for join table
* *ALTER TABLE* gaa4qu*.Subject ADD CONSTRAINT subject\_pk PRIMARY KEY ( ids )* is using for set field ids as primary key for subject table
* *CREATE SEQUENCE gaa4qu.SUBJECT\_id\_SEQ* is using for creating sequence for field id in table Subject. This sequence will start from 1 and will sequence for every new input data into table
* *CREATE OR REPLACE TRIGGER GAA4QU.SUBJECT\_id\_TRG* is using for creating or replace trigger that already exist before for field id in Subject table by command “before insert into Subject table” when field ids in each row table that didn’t have any value or null, this trigger command will begin to create new value for field ids from sequence

Uma imagem com texto, captura de ecrã, software, Ícone de computador

Descrição gerada automaticamente

Figure 1 Subject table

|  |
| --- |
| *CREATE TABLE gaa4qu.Student (*  *Cod\_S INTEGER NOT NULL,*  *name VARCHAR2(50 CHAR),*  *Age INTEGER,*  *Genre VARCHAR2(20 CHAR),*  *grou VARCHAR2(20 CHAR)*  *)*  *PCTFREE 10 PCTUSED 40 TABLESPACE users LOGGING*  *STORAGE ( INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS 2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT )*  *NO INMEMORY;*  *ALTER TABLE gaa4qu.Student ADD CONSTRAINT Student\_pk PRIMARY KEY ( Cod\_S );*  *CREATE SEQUENCE GAA4QU.Student\_Cod\_s\_SEQ*  *START WITH 1*  *NOCACHE*  *ORDER ;*  *CREATE OR REPLACE TRIGGER GAA4QU.Student\_Cod\_s\_TRG*  *BEFORE INSERT ON GAA4QU.Student*  *FOR EACH ROW*  *WHEN (NEW.Cod\_S IS NULL)*  *BEGIN*  *:new.Cod\_S := gaa4qu.Student\_Cod\_s\_seq.nextval;*  *end;*  */* |

*Script for Student table*

* *CREATE TABLE gaa4qu.Student* is using for create new table with name *Student* in gaa4qu account which is my account with field Cod\_S data type integer and set to no null for primary key; *name* data type *varchar2(50 char);age* data type *integer; genre* data type *varchar2(20 char); grou* data type *varchar2(20 char)*.
* *CREATE INDEX gaa4qu.Student\_\_idx ON* is using for creating index for *student* table using field cid\_s so it can be use for join table
* *ALTER TABLE gaa4qu.student ADD CONSTRAINT student\_pk PRIMARY KEY ( Cod\_S )* is using for set field *Cod\_S* as primary key for *Student* table
* *CREATE SEQUENCE gaa4qu. STUDENT\_Cod\_S\_SEQ* is using for creating sequence for field id in *Student* table. This sequence will start from 1 and will sequence for every new input data into table.
* *CREATE OR REPLACE TRIGGER GAA4QU.Student\_Cod\_S\_TRG* is using for creating or replace trigger that already exist before for field *cod\_s* in *student* table by command “before insert into *student* table” when field *Cod\_S* in each row table that didn’t have any value or null, this trigger command will begin to create new value for field Cod\_S from sequence

Uma imagem com texto, captura de ecrã, número, software

Descrição gerada automaticamente

Figure 2 Student table

|  |
| --- |
| *CREATE TABLE gaa4qu.Award (*  *IDA INTEGER NOT NULL,*  *Category VARCHAR2(30 CHAR),*  *ID\_SW INTEGER*  *)*  *PCTFREE 10 PCTUSED 40 TABLESPACE users LOGGING*  *STORAGE ( INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS 2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT )*  *NO INMEMORY;*  *ALTER TABLE gaa4qu.Award ADD CONSTRAINT Award\_pk PRIMARY KEY ( IDA );*  *CREATE SEQUENCE GAA4QU.Award\_IDA\_SEQ*  *START WITH 1*  *NOCACHE*  *ORDER ;*  *CREATE OR REPLACE TRIGGER GAA4QU.Award\_IDA\_TRG*  *BEFORE INSERT ON GAA4QU.Award*  *FOR EACH ROW*  *WHEN (NEW.IDA IS NULL)*  *BEGIN*  *:new.IDA := gaa4qu.Award\_IDA\_seq.nextval;*  *end;*  */* |

Script for award table

* *CREATE TABLE gaa4qu.award* is using for create new table with name games in gaa4qu account which is my account with field *IDA* data type integer and set to no null for primary key; Category data type varchar in size about 30 character; *id\_sw* data type *integer.*
* *CREATE INDEX gaa4qu.award\_\_idx ON* is using for creating index for *award* table using field name so *it* can be use for join table
* *ALTER TABLE gaa4qu.award ADD CONSTRAINT award\_pk PRIMARY KEY ( ida )* is using for set field ida as primary key for games table
* *CREATE SEQUENCE gaa4qu.Award\_ida\_SEQ* is using for creating sequence for field ida in award table. This sequence will start from 1 and will sequence for every new input data into table
* *CREATE OR REPLACE TRIGGER GAA4QU.award\_ida\_TRG* is using for creating or replace trigger that already exist before for field ida in award table by command “before insert into award table” when field ida in each row table that didn’t have any value or null, this trigger command will begin to create new value for field ida from sequence

Table

Description automatically generated

Figure 3 award table

|  |
| --- |
| *CREATE TABLE gaa4qu.Scient\_Work (*  *ID\_SW INTEGER NOT NULL,*  *Title VARCHAR2(20 CHAR),*  *Category CHAR(2),*  *Pag INTEGER*  *)*  *PCTFREE 10 PCTUSED 40 TABLESPACE users LOGGING*  *STORAGE ( INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS 2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT )*  *NO INMEMORY;*  *ALTER TABLE gaa4qu.Scient\_Work ADD CONSTRAINT Scient\_Work\_pk PRIMARY KEY ( ID\_SW );*  *CREATE SEQUENCE GAA4QU.Scient\_Work\_ID\_SW\_SEQ*  *START WITH 1*  *NOCACHE*  *ORDER ;*  *CREATE OR REPLACE TRIGGER GAA4QU.Scient\_Work\_ID\_SW\_TRG*  *BEFORE INSERT ON GAA4QU.Scient\_Work*  *FOR EACH ROW*  *WHEN (NEW.ID\_SW IS NULL)*  *BEGIN*  *:new.ID\_SW := gaa4qu.Scient\_Work\_ID\_SW\_seq.nextval;*  *end;*  */* |

Script for *Sientific\_Work* table

* *CREATE TABLE gaa4qu.Sientific\_Work* is using for create new table with name store in gaa4qu account which is my account with field *id\_sw* data type integer and set to no null for primary key; *title* data type varchar in size about 20 character; *category* data type varchar in size about 2 char; *pag* data type integer.
* *CREATE INDEX gaa4qu. Sientific\_Work\_\_id\_sw ON* is using for creating index for store table using field name so it can be use for join table
* *ALTER TABLE gaa4qu. Sientific\_Work* *ADD CONSTRAINT Sientific\_Work* *\_pk PRIMARY KEY ( id\_sw )* is using for set field id\_sw as primary key for store table
* *CREATE SEQUENCE gaa4qu. Sientific\_Work* *\_id\_sw\_SEQ* is using for creating sequence for field id\_sw in store table. This sequence will start from 1 and will sequence for every new input data into table
* *CREATE OR REPLACE TRIGGER GAA4QU. SIENTIFIC\_WORK\_id\_SW\_TRG* is using for creating or replace trigger that already exist before for field id\_sw in store table by command “before insert into store table” when field id\_sw in each row table that didn’t have any value or null, this trigger command will begin to create new value for field id from sequence

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

Figure 4 Sientific\_Work table

|  |
| --- |
| *CREATE TABLE gaa4qu.Student\_Subject (*  *ID\_SS INTEGER NOT NULL,*  *Cod\_S INTEGER NOT NULL,*  *IDS INTEGER NOT NULL,*  *Grade Integer,*  *FOREIGN key(Cod\_S) REFERENCES Student(Cod\_S),*  *FOREIGN key(IDS) REFERENCES Subject(IDS)*  *)*    *PCTFREE 10 PCTUSED 40 TABLESPACE users LOGGING*  *STORAGE ( INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS 2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT )*  *NO INMEMORY;*  *CREATE INDEX gaa4qu.student\_subject\_idx ON*  *gaa4qu.student\_subject (*  *Grade*  *ASC )*  *TABLESPACE users LOGGING;*    *ALTER TABLE gaa4qu.student\_subject ADD CONSTRAINT student\_subject\_pk PRIMARY KEY ( ID\_SS );*  *CREATE SEQUENCE gaa4qu.student\_subject\_id\_SEQ*  *START WITH 1*  *NOCACHE*  *ORDER ;*  *CREATE OR REPLACE TRIGGER GAA4QU.student\_subject\_id\_TRG*  *BEFORE INSERT ON GAA4QU.student\_subject*  *FOR EACH ROW*  *WHEN (NEW.id\_ss IS NULL)*  *BEGIN*  *:new.id\_ss := gaa4qu.student\_subject\_id\_seq.nextval;*  *end;*  */* |

*Script for Student\_Subject table*

* *CREATE TABLE gaa4qu. Student\_Subject* is using for create new table with name *Student\_Subject* in gaa4qu account which is my account with field *id\_ss* data type integer and set to no null; *grade*: with data type integer; *Cod\_S* data type integer as foreign key with references on Student table; *IDS* data type integer as foreign key with references on Subject table.
* *CREATE INDEX gaa4qu.Student\_Subject \_\_idx ON* is using for creating index for store table using field grade so it can be use for join table
* *ALTER TABLE gaa4qu.pu.Student\_Subject ADD CONSTRAINT Student\_Subject\_pk PRIMARY KEY ( ID\_SS )* is using for set field name as primary key for *. Student\_Subject* table
* *CREATE SEQUENCE gaa4qu.Student\_Subject\_id\_SEQ* is using for creating sequence for field id\_ss in *Student\_Subject* table. This sequence will start from 1 and will sequence for every new input data into table
* *CREATE OR REPLACE TRIGGER GAA4QU.STUDENT\_SUBJECT\_id\_TRG* is using for creating or replace trigger that already exist before for field id\_ss in *. Student\_Subject* table by command “before insert into store table” when field id\_ss in each row table that didn’t have any value or null, this trigger command will begin to create new value for field id\_ss from sequence

Uma imagem com texto, Tipo de letra, software, número

Descrição gerada automaticamente

Figure 5 . Student\_Subject table

|  |
| --- |
| *CREATE TABLE gaa4qu.Student\_work (*  *ID\_SWork INTEGER NOT NULL,*  *ID\_SW INTEGER NOT NULL,*  *Cod\_S INTEGER NOT NULL,*  *Committee VARCHAR2(50 CHAR),*  *FOREIGN key(ID\_SW) REFERENCES SCIENT\_WORK(ID\_SW),*  *FOREIGN key(Cod\_S) REFERENCES STUDENT(Cod\_S)*  *)*  *PCTFREE 10 PCTUSED 40 TABLESPACE users LOGGING*  *STORAGE ( INITIAL 65536 NEXT 1048576 PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS 2147483645 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT )*  *NO INMEMORY;*    *ALTER TABLE gaa4qu.Student\_work ADD CONSTRAINT student\_work\_pk PRIMARY KEY ( ID\_Swork );*  *CREATE SEQUENCE gaa4qu.Student\_work\_id\_SEQ*  *START WITH 1*  *NOCACHE*  *ORDER ;*  *CREATE OR REPLACE TRIGGER GAA4QU.Student\_work\_id\_TRG*  *BEFORE INSERT ON GAA4QU.Student\_work*  *FOR EACH ROW*  *WHEN (NEW.id\_Swork IS NULL)*  *BEGIN*  *:new.id\_swork := gaa4qu.Student\_work\_id\_seq.nextval;*  *end;*  */* |

*Script for Student\_Work table*

* *CREATE TABLE gaa4qu.Student\_Work* is using for create new table with name games in gaa4qu account which is my account with field *id\_Swork* data type integer and set to no null for primary key; *Cod\_S* data type integer as foreign key with reference on Student table; *ID\_SW* data type integer as foreign key with reference on *Scientific\_Work* table and field Committee data type varchar2 in size about 50 character.
* *CREATE INDEX gaa4qu. Scientific\_Work*\_*idx ON* is using for creating index for games table using field name so it can be use for join table
* *ALTER TABLE gaa4qu.Scientific\_Work ADD CONSTRAINT Scientific\_Work\_pk PRIMARY KEY ( ID\_SWork )* is using for set field id as primary key for games table
* *CREATE SEQUENCE gaa4qu. Scientific\_Work\_id\_SEQ* is using for creating sequence for field id\_Swork in games table. This sequence will start from 1 and will sequence for every new input data into table
* *CREATE OR REPLACE TRIGGER GAA4QU. Scientific\_Work\_id\_TRG* is using for creating or replace trigger that already exist before for field id\_swork in *Scientific\_Work* table by command “before insert into games table” when field id\_swork in each row table that didn’t have any value or null, this trigger command will begin to create new value for field id from sequence

Uma imagem com texto, Tipo de letra, número, software

Descrição gerada automaticamente

Figure 6 Scientific\_Work table

* 1. **Load Data**

After I create all of the 6 tables, then I load my data into the tables with this step:

1. Uma imagem com texto, captura de ecrã, software, Ícone de computador

   Descrição gerada automaticamenteRight click on the table name, in this example I use *Student* table. After right click then will be show up some menu and then click Import Data
2. Data Import wizard will show up, choose your file by click browse menu. Click file name which is *Student.csv* and click open. After that click next go to Import Method

Uma imagem com texto, captura de ecrã, software, ecrã

Descrição gerada automaticamente

1. In import method you will see you data in there, let others and click next. Go to Choose Columns

Uma imagem com texto, captura de ecrã, software, ecrã

Descrição gerada automaticamente

1. In Choose Columns let the others and click next again. Go to Column Definition

Uma imagem com texto, captura de ecrã, software, ecrã

Descrição gerada automaticamente

Graphical user interface, text, application, Word

Description automatically generated

1. Importing data will be process

Graphical user interface, text, application, email

Description automatically generated

1. If success import data, there are a notification will show up

Graphical user interface, text, application, email

Description automatically generated

I repeat this process 5 times to load all of my data in 5 another tables into database. Which is Subject, Student\_Subject, Student\_Work, and Scientific\_Work and Award table. Also I include script shell for data load for all of my data in this submit assignment.

1. **Data model Image and Relational Data Model**

Uma imagem com texto, file, diagrama, captura de ecrã

Descrição gerada automaticamente

The image above was using *Draw.oi* application is Data Model Image and Relational Data Model. Here I will explain each table and the relation that they have: ***Student*** table have 4 field:

- CoD\_S: with data type integer and as a primary key for the table and also I put CoD\_S into index so it can use for relation with other table; Name: with data type varchar in size about 50 characters; Age: with data type Integer; Genre: with data type varchar in size about 30; Group: with data type varchar in size about 1.

**Subject** table have 2 field:

- ID\_S: with data type integer and as a primary key for the table - name: with data type varchar in size about 30 character.

**Student\_Subject** table have 4 field: - ID\_SS: with data type integer and as a primary key for the table – Cod\_S: with data type integer and I set it as a foreign key comes from Student table; IDS: with data type integer and I set it as a foreign key comes from Subject table and field Grade: with data type integer (1-5).

**Student\_Work** table have 3 field: - ID\_SWork: with data type integer and as a primary key for the table – Cod\_S: with data type integer and I set it as a foreign key comes from Student table; ID\_SW: with data type integer and I set it as a foreign key comes from Scient\_Work table and field Committee: with data type varchar in size about 50 character.

**Scientific\_Work** table have 4 field: - ID\_SW: with data type integer and as a primary key for the table and also I put ID\_SW into index so it can use for relation with other table; Title: with data type varchar in size about 50 characters; Page: with data type Integer and Category: with data type varchar in size about 1.

**Award** table have 2 field: - ID\_A: with data type integer and as a primary key for the table and also I put ID\_A into index so it can use for relation with other table; Category: with data type varchar in size about 20 characters; ID\_SW: with data type Integer, I set it as a foreign key comes from Student table.

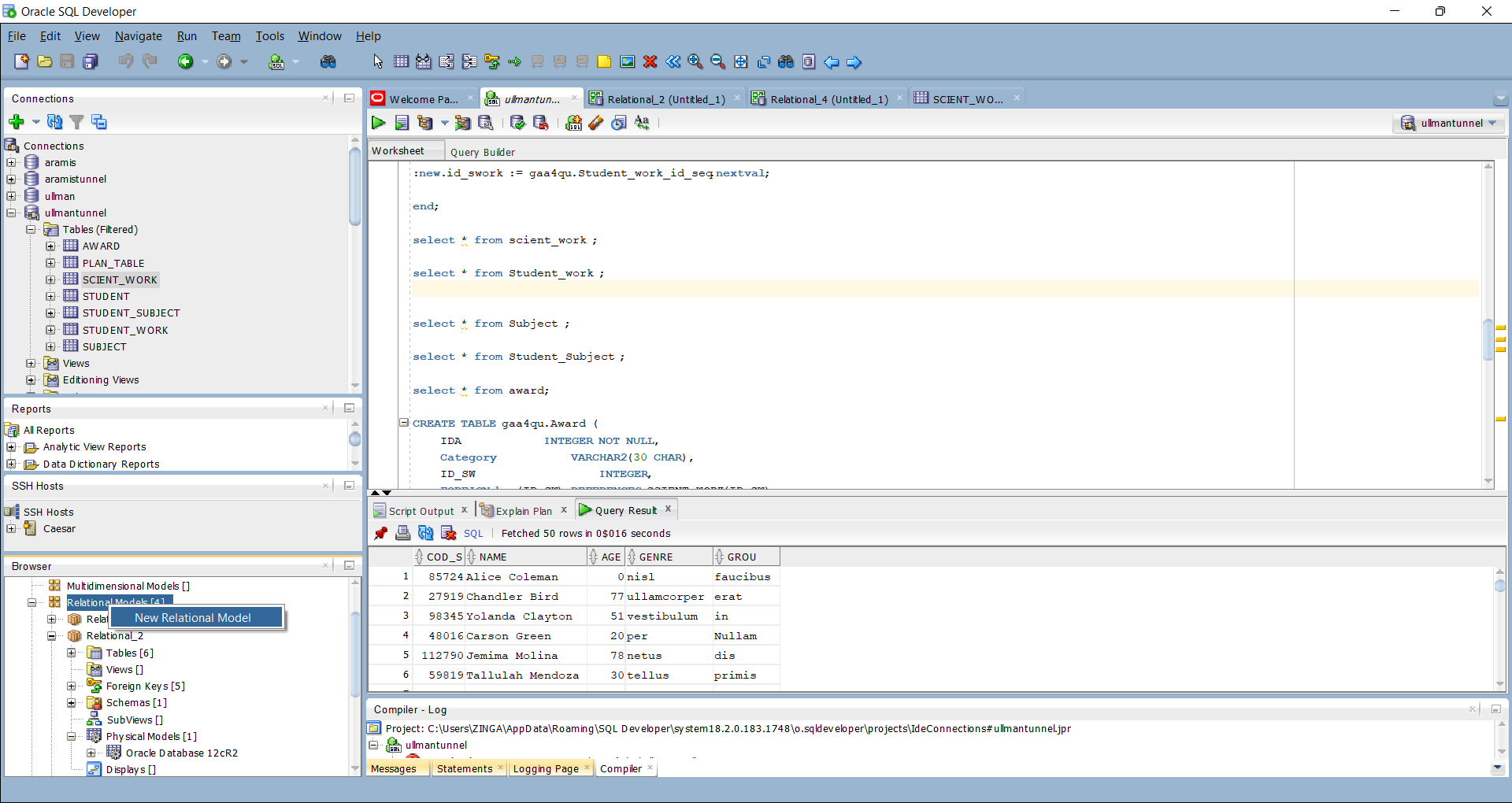
**However I can also generate relational data model on SQL Developer with the steps:**

1. Clicking on: *View* - > Data Modeler - > Browser

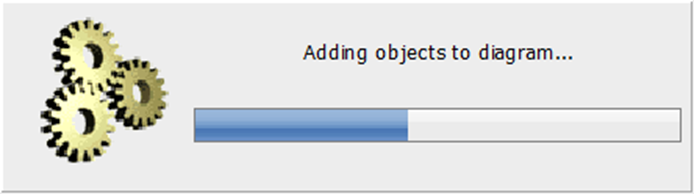
Uma imagem com texto, captura de ecrã, software, Ícone de computador

Descrição gerada automaticamente

1. After click on: New Relational Model



1. Once the windows of new relation Model is open, I drag the table *Student* on the that window:



1. And the table *Student* luck like this:Uma imagem com texto, captura de ecrã, software, ecrã

   Descrição gerada automaticamente
2. Uma imagem com texto, captura de ecrã, ecrã, software

   Descrição gerada automaticamenteAfter dragging all table the scheme luck like this
3. **Project Task**

**View 1:**

**I create a view with name *all\_students\_subect\_with\_latters\_a\_z* wich select all IDS and Names of subject that was given to the students with latters from A – Z.**

**CREATE VIEW** ALL\_STUDENTS\_SUBECT\_WITH\_LATTERS\_A\_Z (ID\_SUBJECT, NAME\_SUBJECT) **AS**

SELECT SUB.IDS, SUB.NAME

**FROM** STUDENT STU JOIN STUDENT\_SUBJECT SSB ON STU.COD\_S=SSB.COD\_S

JOIN SUBJECT SUB ON SSB.IDS = SUB.IDS

**WHERE** REGEXP\_LIKE(SUB.NAME, '[A-ZA-Z]') ;

After creating the view, I call the view with this code:

**SELECT** \* FROM ALL\_STUDENTS\_SUBECT\_WITH\_LATTERS\_A\_Z;

And I had this result table:

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

This View show all *IDS* and *Names* of subject that was given to the students with latters from A – Z. For that I Showed field ids using subject.ids and subject.name with joining table Subject and Student\_Subject using parameter field ids in Subject tabel have same value with field ids in Student\_Subject, and then joining again with Student table using parameter field Cod\_S in Student and match all strings that start with a letter on table Subject.

**VIEW 2:**

**Return the number of students per subject using the having count this number must be less than 2.**

**CREATE VIEW** SUBJECT\_PER\_QUANTITY (SUBJECT\_NAME,STUDENT\_QUANTITY ) **AS**

**SELECT** SUBJECT.NAME , COUNT(STUDENT\_SUBJECT.COD\_S)

**FROM** STUDENT JOIN STUDENT\_SUBJECT ON STUDENT.COD\_S=STUDENT\_SUBJECT.COD\_S JOIN SUBJECT ON STUDENT\_SUBJECT.IDS=SUBJECT.IDS

**GROUP BY** SUBJECT.NAME

**HAVING** COUNT(STUDENT\_SUBJECT.COD\_S) < 2;

After creating the view, I call the view with this code:

**SELECT** \* FROM SUBJECT\_PER\_QUANTITY;

And I had this result table:

Uma imagem com texto, Tipo de letra, file, número

Descrição gerada automaticamente

This second View show Return the number of students per subject using the having count this number must be less than 2. For that I Joined the tables Student and Student.Cod\_S with fiel Cod\_S and I also joining the table Student\_Subject and Subject using parameter field IDS, I used the function Group by name of subject and having count less than 2 quantity of students.

create or replace function PAGE\_avg(n integer)

return number

is

cursor curs1 is select Scient\_work.ID\_SW, SCIENT\_WORK.TITLE, SCIENT\_WORK.CATEGORY, SCIENT\_WORK.PAG

from SCIENT\_WORK

where SCIENT\_WORK.PAG = n;

rec curs1%ROWTYPE;

s float;

c int;

BEGIN

s := 0.0;

c := 0;

open curs1;

LOOP

FETCH curs1 INTO rec;

EXIT WHEN curs1%NOTFOUND;

c := c + 1;

s := s + rec.PAG;

END LOOP;

CLOSE curs1;

return round((s/c),2);

END;

CREATE OR REPLACE TRIGGER ID\_TRIGGER

AFTER INSERT OR UPDATE ON AWARD

FOR EACH ROW

DECLARE

C INTEGER;

BEGIN

SELECT COUNT(\*) INTO C FROM AWARD WHERE IDA = :NEW.IDA;

IF C = 0 THEN

INSERT INTO AWARD(IDA) VALUES(:NEW.IDA);

DBMS\_OUTPUT.PUT\_LINE('NEW AWARD SUCESSFUL INSERTED ');

END IF;

END;

CREATE OR REPLACE TRIGGER ID\_SUBJECT\_TRIGGER

AFTER INSERT OR UPDATE ON SUBJECT

FOR EACH ROW

DECLARE

C INTEGER;

BEGIN

SELECT COUNT(\*) INTO C FROM SUBJECT WHERE IDS = :NEW.IDS;

IF C = 0 THEN

INSERT INTO SUBJECT(IDS) VALUES(:NEW.IDS);

DBMS\_OUTPUT.PUT\_LINE('NEW SUBJECT INSERTED TOO');

END IF;

END;

**CREATE OR REPLACE VIEW** PASS\_OR\_FAIL AS

**SELECT** STUDENT.COD\_S, STUDENT.NAME , STUDENT.AGE, STUDENT.GENRE, STUDENT.GROU

**FROM** STUDENT JOIN STUDENT\_SUBJECT ON STUDENT.COD\_S= STUDENT\_SUBJECT.COD\_S

**WHERE** STUDENT\_SUBJECT.GRADE BETWEEN 2 AND 5;