جامعة جدة
University of Jeddah

Database 18987

Project title: The Impact of Vaccine on Elementary School Students

Team members: three

Student names:

Sadeem Alsawat – 2009293

Dania Al-Ashour -1915078

Amjaad Hasan – 2007936

Email:

2009293@uj.edu.sa

Dashour.stu@uj.edu.sa

2007936@uj.edu.sa

Instructor Name:

Bushra Alanesi

Eman Almadhon

Section: A3-A3L

Submission date: 21/5/2022

Table of contents

Project Description

This project will be more like a statistic of an elementary school that describes the number of students from the age of 5 to 12 that have or have not taken the vaccine for many reasons for example, They cannot take it for medical reasons such as respiratory diseases and parents' disapproval because of their young age and what is the result of the spread of COVID-19 between vaccinated and non-vaccinated students in the event of a child being infected in the classroom. presenting the difference between children vaccinated and children non-vaccinated, and which age group took vaccines more than others.

There are also some laws related to the COVID-19 vaccine for children:

- Children between 5-12 are allowed to take the vaccine but it is not mandatory.

- Children dose is less than the olders because they are still developing.

- Children's family members are not obligated to be vaccinated to attend their schools.

Scope of Database

The project will contain Entities called: STUDENT that has the students (name, ID, age, grade level, group A or B), DOSE (dose name, the number of dose, dose ID, and what's the side effects of it), VACCINE (vaccine ID, vaccine name, and period between dose), REASON (reason ID, reason type, reason description)

What are the output of the system

The school administration will be able to view all records of students who received the COVID-19 vaccine and those who did not receive it with reasons, whether legal such as age or diseases, or illegal such as parental disapproval.

Entity Description Table

Entity	Description	PR
Student	A table that holds information about students.	ID
Reason	A with reasons for not vaccinating students.	Reason_ID
Vaccine	A table that holds information about vaccine.	Vaccine_ID
dose	A table that holds information about doses connected with vaccine table and represent more information.	Dose_ID

Attribute Description Table

Student table:

ID	Simple primary key	Contain student ID.
Name	composite	Contain the first and last name of the student.
Age	simple	Contain the age of the student.
Grade	simple	Contain the grade of the student are in.
Group	simple	Contain which group of the student are in.

Vaccine table:

Vaccine ID	Simple primary key	Contain Vaccine ID.
Vaccine name	simple	Contain the name of the vaccine.
Period	simple	Contain the time of the vaccine it takes.

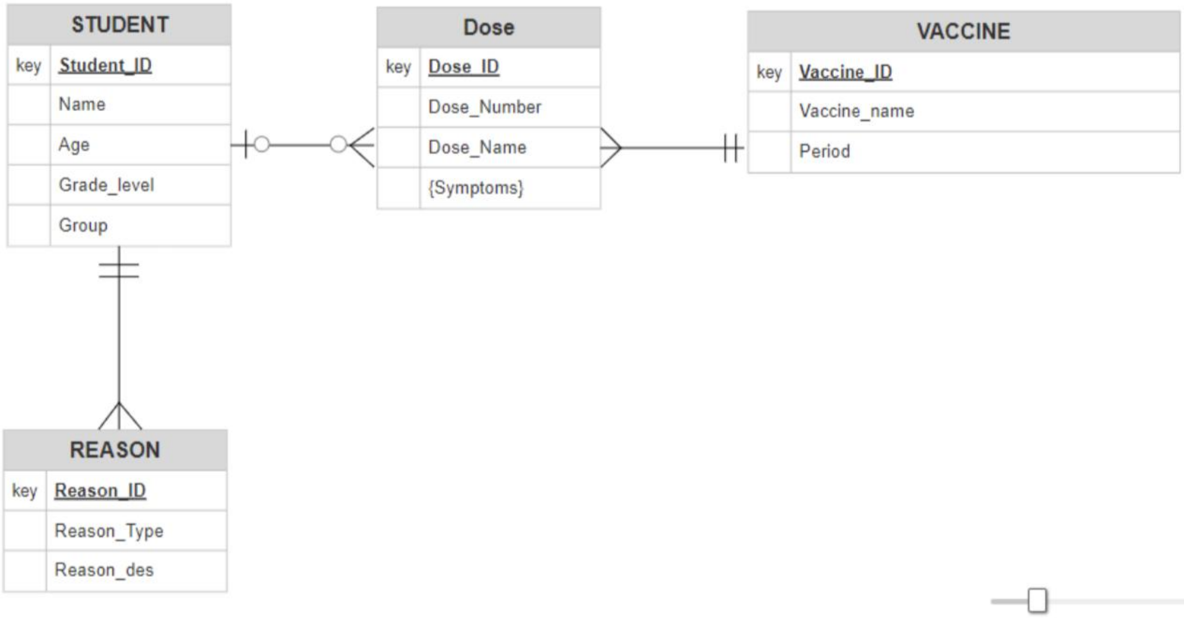
Dose table:

Dose ID	Simple primary key	Contain Dose ID.
Dose number	Simple	Contain the number of the doses that student takes it.
Dose name	Simple	Contain the name of the dose.
Symptoms	Multivalued	Contain the Symptoms of the student that comes because of dose vaccine.

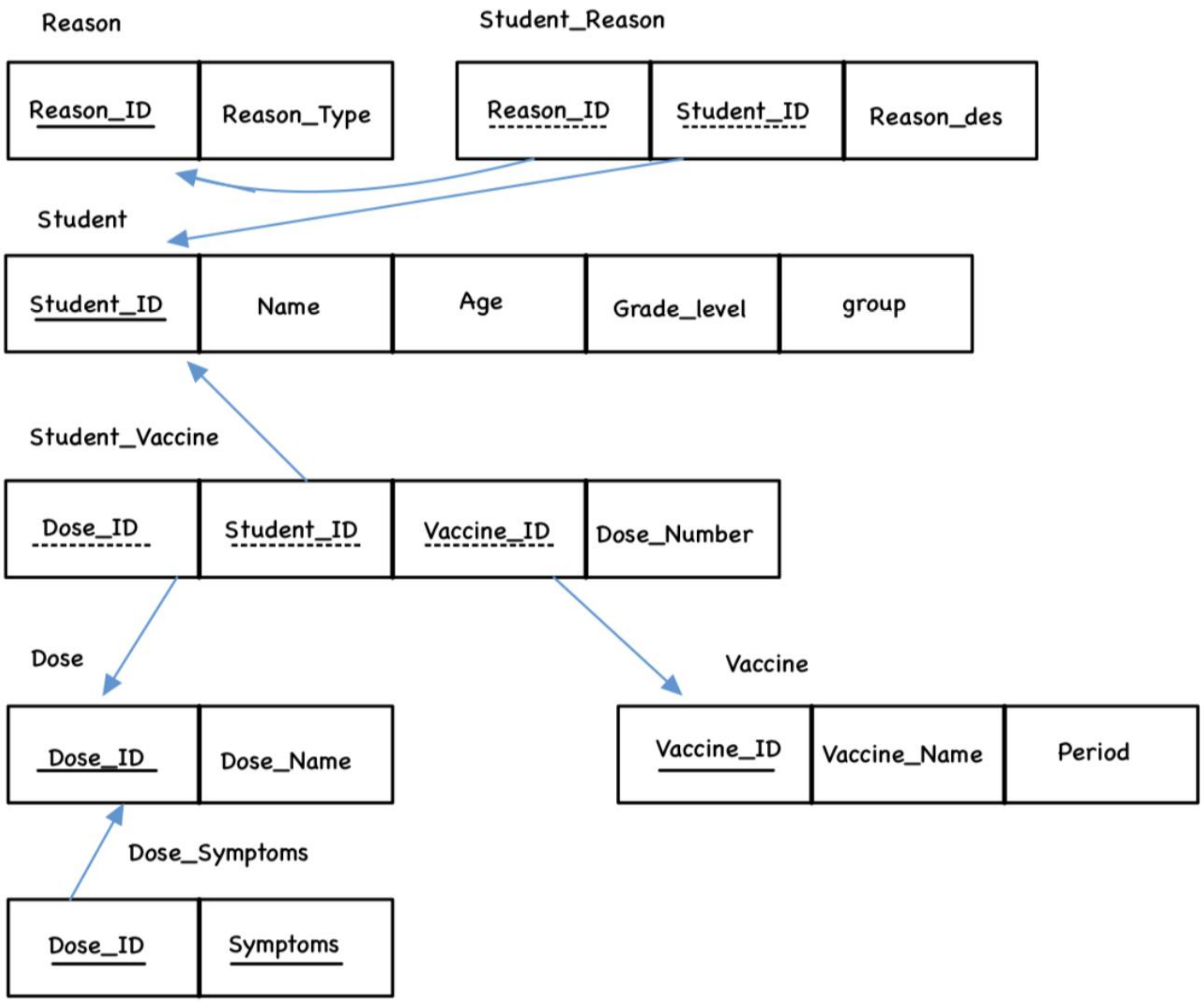
Reason table:

Reason ID	Simple primary key	Contain Reason ID.
Reason type	Multivalued	Contain the type of reasons for not taking the vaccine.
Reason_des	Simple	Contain the description of the reason for not taking the vaccine.

ER Diagram


STUDENT
key
Student_ID
Name
Age
Grade_level
Group
Dose
key
Dose_ID
Dose_Number
Dose_Name
{Symptoms}
VACCINE
key
Vaccine_ID
Vaccine_name
Period
REASON
key
Reason_ID
Reason_Type
Reason_des

Relational


Reason
Reason_ID
Reason_Type
Student_Reason
Reason_ID
Student_ID
Reason_des
Student
Student_ID
Name
Age
Grade_level
group
Student_Vaccine
Dose_ID
Student_ID
Vaccine_ID
Dose_Number
Dose
Dose_ID
Dose_Name
Vaccine
Vaccine_ID
Vaccine_Name
Period
Dose_Symptoms
Dose_ID
Symptoms

Functional dependencies

#Student_ID → Name, Age, gradeLevel, group,

#Reason_ID → Reason_type , Reason_description

#Vaccine_ID → Vaccine_name, Period

#SymptomsID → Description

#Dose_ID → Dose_number, Dose_name

Normalization

Student (Student_Id, Name, Age, Grade-Level, Group) in 3NF

there are no repeating groups, and a unique key has been identified for this relation.

all non-key attributes are fully functionally dependent on the entire key and

there's no Transitive dependency.

Reason (Reason_Id, Student_Id ,Reason_Type, Reason_des) in 1NF

there are no repeating groups, and a unique key has been identified for this relation.

The 2NF is to remove partial dependency.

Reason (Reason_Id, Reason_Type) in 3NF

Student_Reason (Reason_Id, Student_Id , Reason_des) in 3NF

all non-key attributes are fully functionally dependent on the entire key and

there's no Transitive dependency.

Vaccine (Vaccine_Id, Vaccine_name, period) in 3NF

there are no repeating groups, and a unique key has been identified for this relation.

all non-key attributes are fully functionally dependent on the entire key and

there's no Transitive dependency.

Dose (Dose_Id, Student_Id, Vaccine_Id, Dose_Number, Dose_Name) in 1NF

there are no repeating groups, and a unique key has been identified for this relation.

The 2NF is to remove partial dependency.

Student_Vaccinated (Dose_Id, Student_Id, Vaccine_Id, Dose_Number) in 3NF

Dose (Dose_Id, Dose_Name) in 3NF

all non-key attributes are fully functionally dependent on the entire key and

there's no Transitive dependency.

Dose_Symptoms (Dose_Id, Symptoms) in 3NF

there are no repeating groups, and a unique key has been identified for this relation.

all non-key attributes are fully functionally dependent on the entire key and

there's no Transitive dependency.

Anomalies of NF1:

Insertion: cannot insert a dose details until the student gets the vaccine.

Deletion: if student is deleted all details of the vaccine and dose will be deleted as well.

Update: if a detail of a specific vaccine needs to be updated, information of the students who took the vaccine needs to be updated.

Anomalies of NF2:

Insertion: a new student cannot be inserted unless they have taken the vaccine or have a reason for not taking it.

Deletion: if a vaccine is deleted all details of the student and dose will be deleted as well.

Update: if a detail of a specific student needs to be updated, all doses or reasons for that student needs to be updated.

All Tables After Normalization in 3NF:

Student (Student_Id, Name, Age, Grade-Level, Group)

Reason (Reason_Id, Reason_Type)

Student_Reason (Reason_Id, Student_Id , Reason_des)

Vaccine (Vaccine_Id, Vaccine_name, period)

Student_Vaccinated (Dose_Id, Student_Id, Vaccine_Id, Dose_Number)

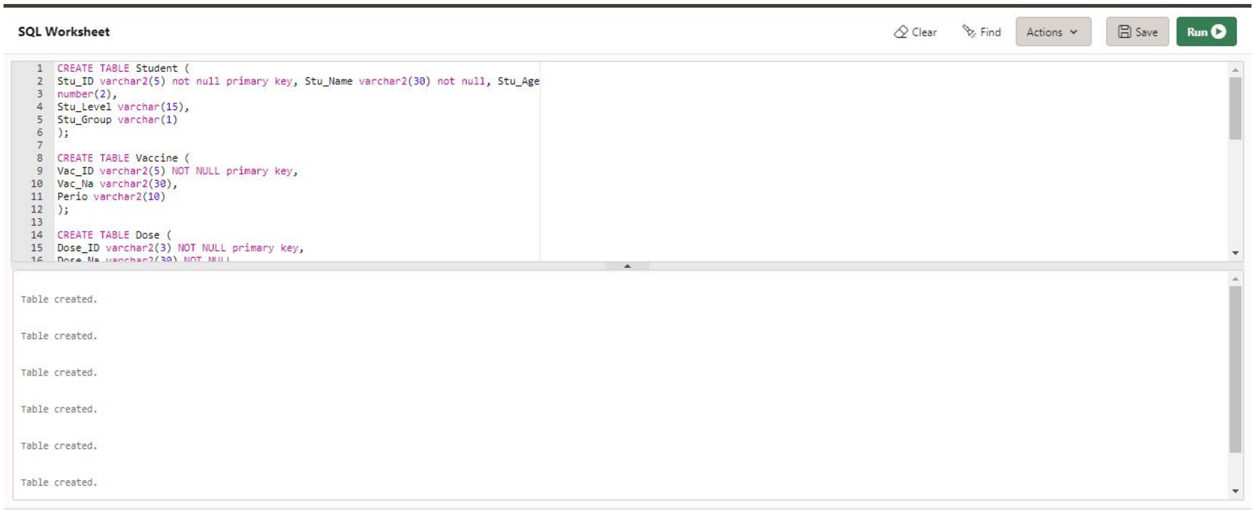
Dose (Dose_Id, Dose_Name)

Dose_Symptoms (Dose_Id, Symptoms)

Schema

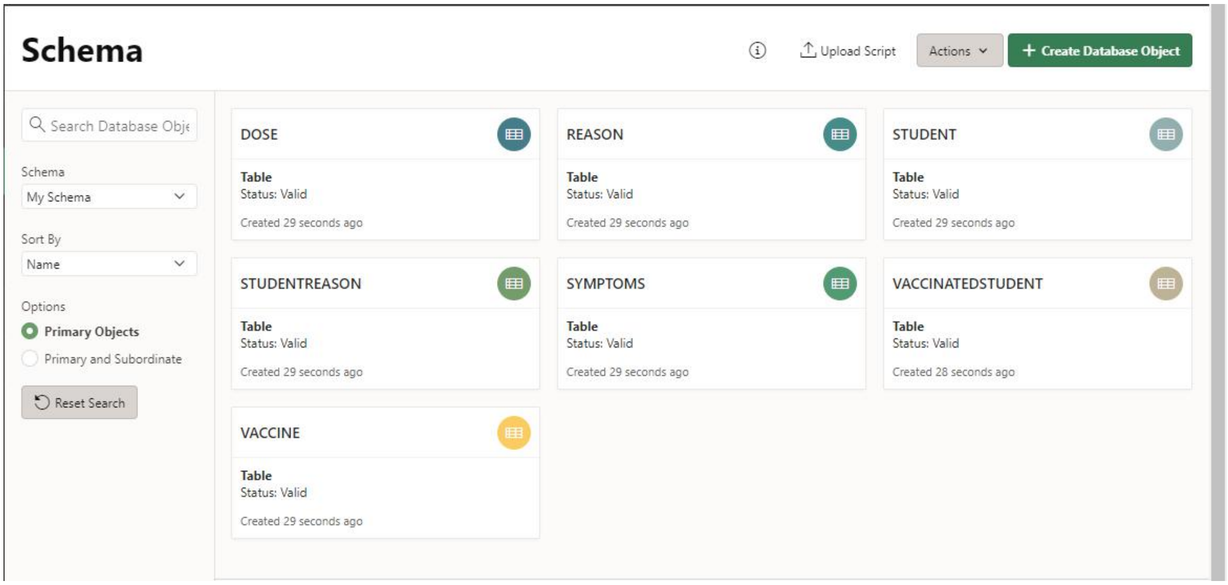
Table	Column name	Data type	Constraint
Student	Stu_ID	Varchar2 (5)	Primary key
	Stu_name	Varchar2 (30)	Not null
	Stu_age	Number (2)	
	Stu_level	Varchar2 (1)	
	Stu_Groub	Varchar2 (1)	
Vaccine	Vac_ID	Varchar2 (5)	Primary key
	Vac_name	Varchar2 (30)	
	perio	Varchar2 (10)	
Dose	Dose_ID	Varchar2 (3)	Primary key
	Dose_name	Varchar2 (30)	Not null
Symptoms	S_ID	Number (4)	Primary key
	Dose_ID	Varchar2 (3)	Foreign key
	Description	Varchar2 (100)	
Reason	Rsn_ID	Varchar2 (5)	Primary key
	Rsn_type	Varchar2 (30)	
StudentReason	Rsn_ID	Varchar2 (5)	Foreign key
	Rsn_dec	Varchar2 (50)	
	Stu_ID	Varchar2 (5)	Foreign key
VaccinatedStudent	Dose_ID	Varchar2 (3)	Foreign key
	Dose_no	Number (3)	
	Stu_ID	Varchar2 (5)	Foreign key
	Vac_ID	Varchar2 (5)	Foreign key

Final Resulting Table



```
CREATE TABLE Student (
Stu_ID varchar2(5) not null primary key, Stu_Name varchar2(30) not null, Stu_Age
number(2),
Stu_Level varchar(15),
Stu_Group varchar(1)
);

CREATE TABLE Vaccine (
Vac_ID varchar2(5) NOT NULL primary key,
Vac_Na varchar2(30),
Perio varchar2(10)
);

CREATE TABLE Dose (
Dose_ID varchar2(3) NOT NULL primary key,
Dose_Na varchar2(30) NOT NULL
```

```
Table created.

Table created.

Table created.

Table created.

Table created.

Table created.
```



DOSE
Table
Status: Valid
Created 29 seconds ago

REASON
Table
Status: Valid
Created 29 seconds ago

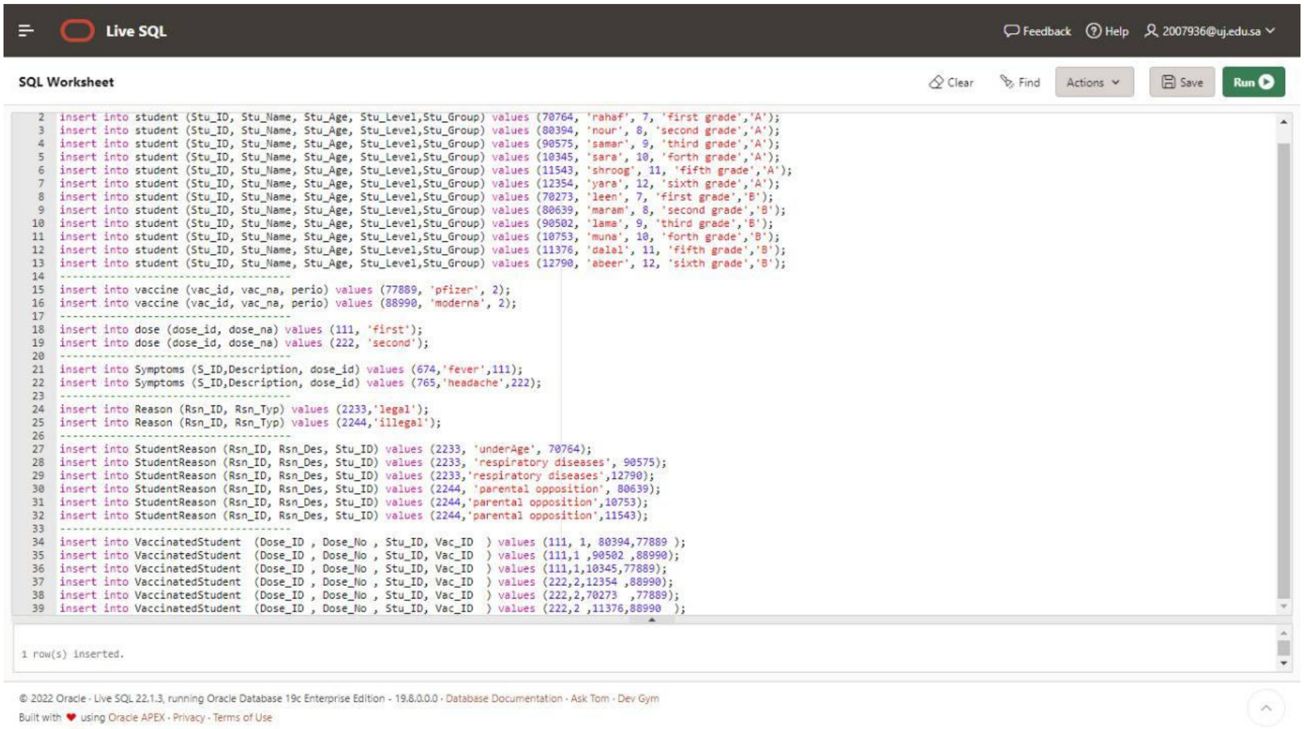
STUDENT
Table
Status: Valid
Created 29 seconds ago

STUDENTREASON
Table
Status: Valid
Created 29 seconds ago

SYMPTOMS
Table
Status: Valid
Created 29 seconds ago

VACCINATEDSTUDENT
Table
Status: Valid
Created 28 seconds ago

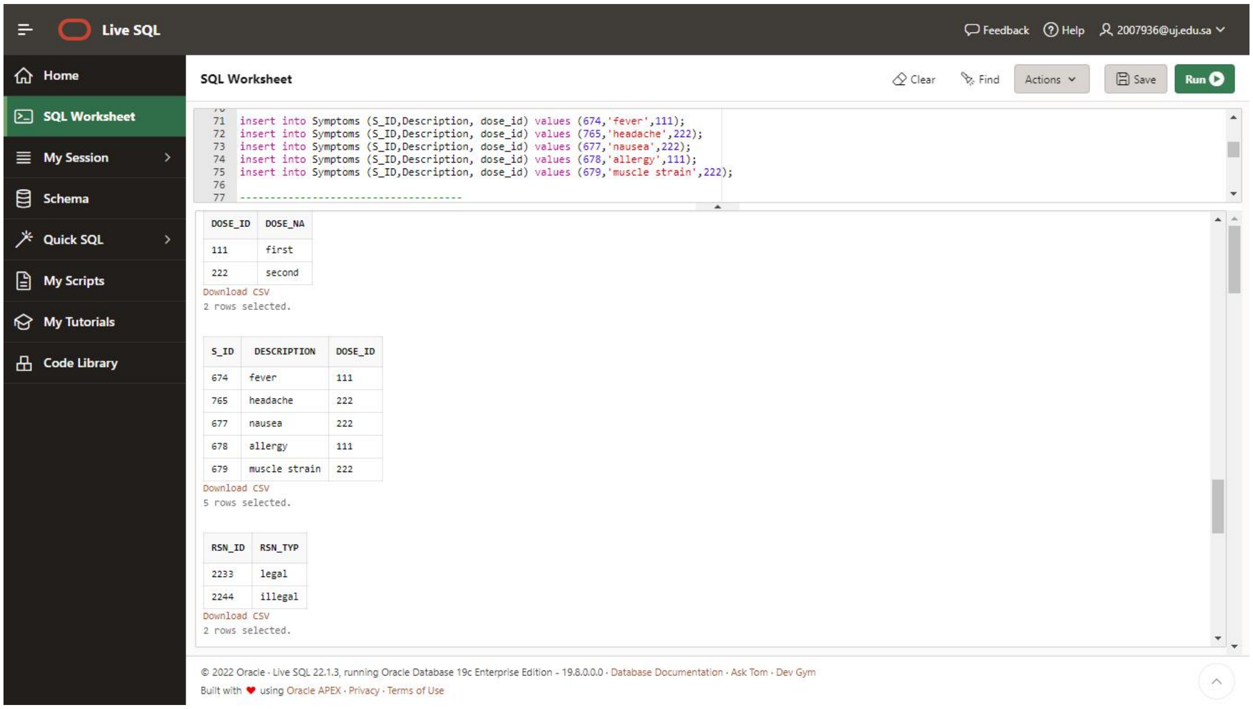
VACCINE
Table
Status: Valid
Created 29 seconds ago



```sql
insert into student (Stu_ID, Stu_Name, Stu_Age, Stu_Level,Stu_Group) values (70764, 'rahaf', 7, 'first grade','A');
insert into student (Stu_ID, Stu_Name, Stu_Age, Stu_Level,Stu_Group) values (80394, 'nour', 8, 'second grade','A');
insert into student (Stu_ID, Stu_Name, Stu_Age, Stu_Level,Stu_Group) values (90575, 'samar', 9, 'third grade','A');
insert into student (Stu_ID, Stu_Name, Stu_Age, Stu_Level,Stu_Group) values (10345, 'sara', 10, 'forth grade','A');
insert into student (Stu_ID, Stu_Name, Stu_Age, Stu_Level,Stu_Group) values (11543, 'shroog', 11, 'fifth grade','A');
insert into student (Stu_ID, Stu_Name, Stu_Age, Stu_Level,Stu_Group) values (12354, 'yara', 12, 'sixth grade','A');
insert into student (Stu_ID, Stu_Name, Stu_Age, Stu_Level,Stu_Group) values (70273, 'leen', 7, 'first grade','B');
insert into student (Stu_ID, Stu_Name, Stu_Age, Stu_Level,Stu_Group) values (80639, 'maram', 8, 'second grade','B');
insert into student (Stu_ID, Stu_Name, Stu_Age, Stu_Level,Stu_Group) values (90502, 'lama', 9, 'third grade','B');
insert into student (Stu_ID, Stu_Name, Stu_Age, Stu_Level,Stu_Group) values (10753, 'muna', 10, 'forth grade','B');
insert into student (Stu_ID, Stu_Name, Stu_Age, Stu_Level,Stu_Group) values (11376, 'dalal', 11, 'fifth grade','B');
insert into student (Stu_ID, Stu_Name, Stu_Age, Stu_Level,Stu_Group) values (12790, 'abeer', 12, 'sixth grade','B');
-------------------------------------
insert into vaccine (vac_id, vac_na, perio) values (77889, 'pfizer', 2);
insert into vaccine (vac_id, vac_na, perio) values (88990, 'moderna', 2);
-------------------------------------
insert into dose (dose_id, dose_na) values (111, 'first');
insert into dose (dose_id, dose_na) values (222, 'second');
-------------------------------------
insert into Symptoms (S_ID,Description, dose_id) values (674,'fever',111);
insert into Symptoms (S_ID,Description, dose_id) values (765,'headache',222);
-------------------------------------
insert into Reason (Rsn_ID, Rsn_Typ) values (2233,'legal');
insert into Reason (Rsn_ID, Rsn_Typ) values (2244,'illegal');
-------------------------------------
insert into StudentReason (Rsn_ID, Rsn_Des, Stu_ID) values (2233, 'underAge', 70764);
insert into StudentReason (Rsn_ID, Rsn_Des, Stu_ID) values (2233, 'respiratory diseases', 90575);
insert into StudentReason (Rsn_ID, Rsn_Des, Stu_ID) values (2233,'respiratory diseases',12790);
insert into StudentReason (Rsn_ID, Rsn_Des, Stu_ID) values (2244, 'parental opposition', 80639);
insert into StudentReason (Rsn_ID, Rsn_Des, Stu_ID) values (2244,'parental opposition',10753);
insert into StudentReason (Rsn_ID, Rsn_Des, Stu_ID) values (2244,'parental opposition',11543);
-------------------------------------
insert into VaccinatedStudent  (Dose_ID , Dose_No , Stu_ID, Vac_ID  ) values (111, 1, 80394,77889 );
insert into VaccinatedStudent  (Dose_ID , Dose_No , Stu_ID, Vac_ID  ) values (111,1 ,90502 ,88990);
insert into VaccinatedStudent  (Dose_ID , Dose_No , Stu_ID, Vac_ID  ) values (111,1,10345,77889);
insert into VaccinatedStudent  (Dose_ID , Dose_No , Stu_ID, Vac_ID  ) values (222,2,12354 ,88990);
insert into VaccinatedStudent  (Dose_ID , Dose_No , Stu_ID, Vac_ID  ) values (222,2,70273  ,77889);
insert into VaccinatedStudent  (Dose_ID , Dose_No , Stu_ID, Vac_ID  ) values (222,2 ,11376,88990  );
```

```
1 row(s) inserted.
```

© 2022 Oracle · Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use


Live SQL
Feedback
Help
2007936@uj.edu.sa
Home
SQL Worksheet
My Session
Schema
Quick SQL
My Scripts
My Tutorials
Code Library
SQL Worksheet
Clear
Find
Actions
Save
Run

```sql
insert into Symptoms (S_ID,Description, dose_id) values (674,'fever',111);
insert into Symptoms (S_ID,Description, dose_id) values (765,'headache',222);
insert into Symptoms (S_ID,Description, dose_id) values (677,'nausea',222);
insert into Symptoms (S_ID,Description, dose_id) values (678,'allergy',111);
insert into Symptoms (S_ID,Description, dose_id) values (679,'muscle strain',222);

-------------------------------------
```

DOSE_ID	DOSE_NA
111	first
222	second

Download CSV
2 rows selected.

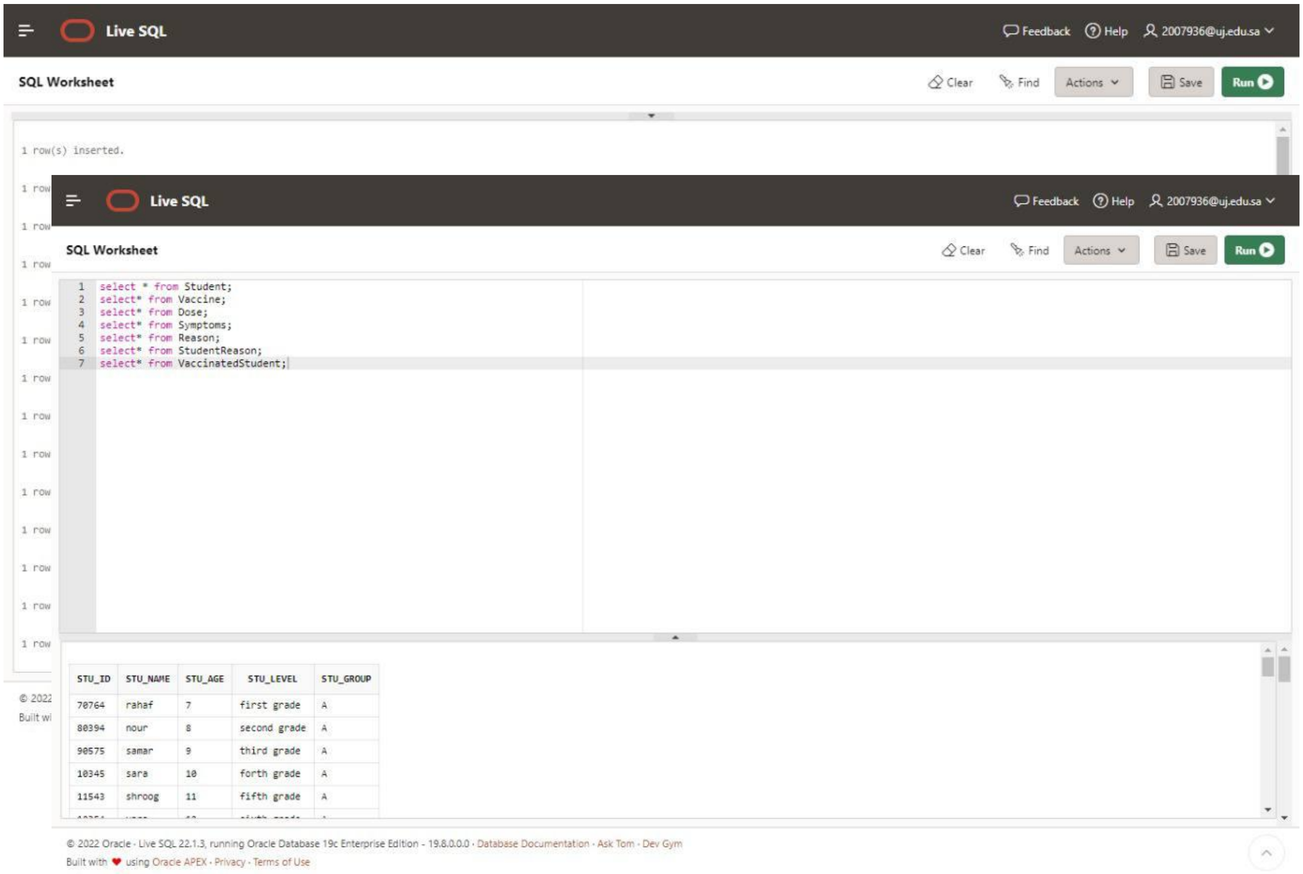
S_ID	DESCRIPTION	DOSE_ID
674	fever	111
765	headache	222
677	nausea	222
678	allergy	111
679	muscle strain	222

Download CSV
5 rows selected.

RSN_ID	RSN_TYP
2233	legal
2244	illegal

Download CSV
2 rows selected.

© 2022 Oracle · Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

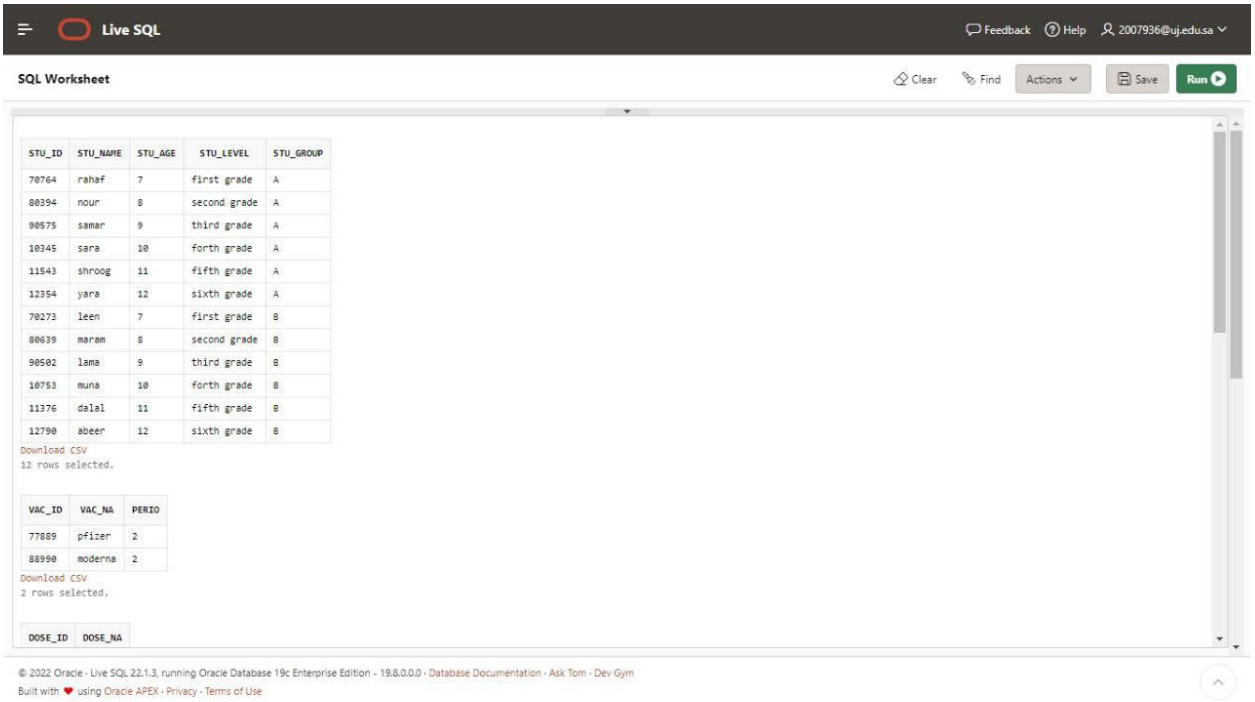


```
1 row(s) inserted.

1 row
1 row
1 row
1 row
1 row
1 row
1 row
1 row
1 row
1 row
1 row
1 row
1 row
```

```
select * from Student;
select* from Vaccine;
select* from Dose;
select* from Symptoms;
select* from Reason;
select* from StudentReason;
select* from VaccinatedStudent;
```

STU_ID	STU_NAME	STU_AGE	STU_LEVEL	STU_GROUP
70764	rahaf	7	first grade	A
80394	nour	8	second grade	A
90575	samar	9	third grade	A
10345	sara	10	forth grade	A
11543	shroog	11	fifth grade	A

© 2022 Oracle · Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ❤ using Oracle APEX · Privacy · Terms of Use



STU_ID	STU_NAME	STU_AGE	STU_LEVEL	STU_GROUP
70764	rahaf	7	first grade	A
80394	nour	8	second grade	A
90575	samar	9	third grade	A
10345	sara	10	forth grade	A
11543	shroog	11	fifth grade	A
12354	yara	12	sixth grade	A
70273	leen	7	first grade	B
80639	maram	8	second grade	B
90502	lama	9	third grade	B
10753	muna	10	forth grade	B
11376	dalal	11	fifth grade	B
12790	abeer	12	sixth grade	B

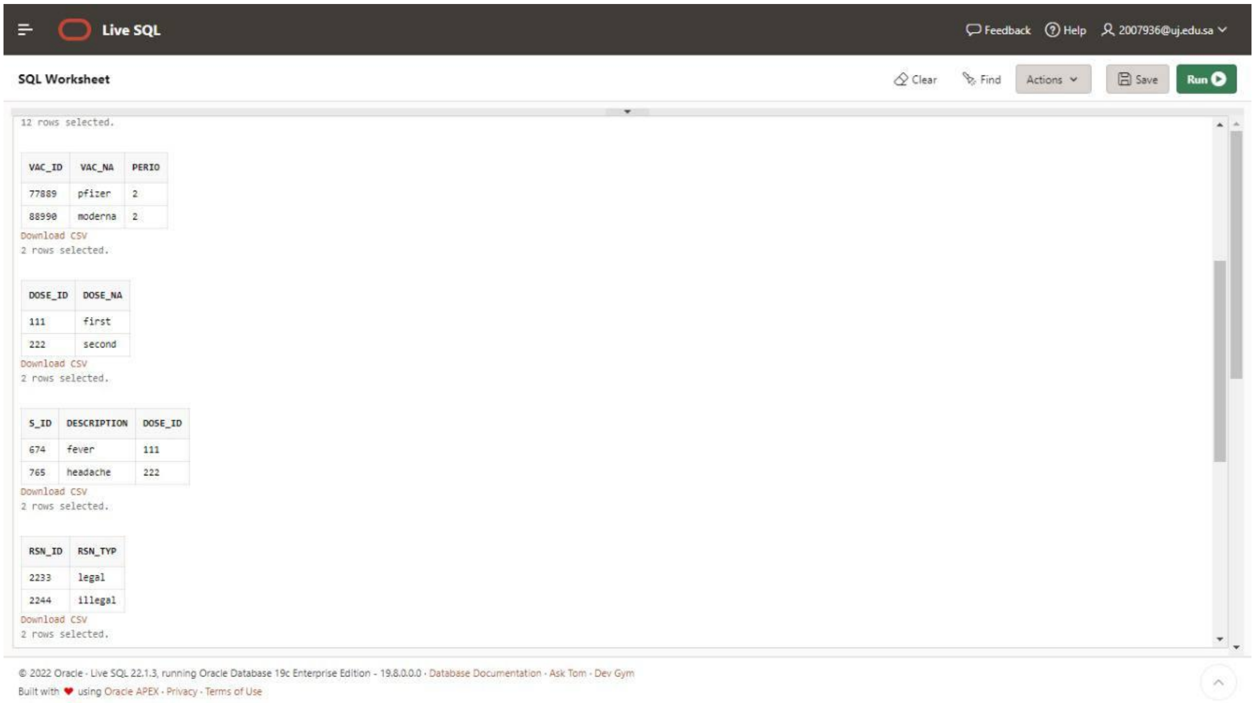
```
Download CSV
12 rows selected.
```

VAC_ID	VAC_NA	PERIO
77889	pfizer	2
88990	moderna	2

```
Download CSV
2 rows selected.
```

DOSE_ID	DOSE_NA

© 2022 Oracle · Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use



```
12 rows selected.
```

VAC_ID	VAC_NA	PERIO
77889	pfizer	2
88990	moderna	2

```
Download CSV
2 rows selected.
```

DOSE_ID	DOSE_NA
111	first
222	second

```
Download CSV
2 rows selected.
```

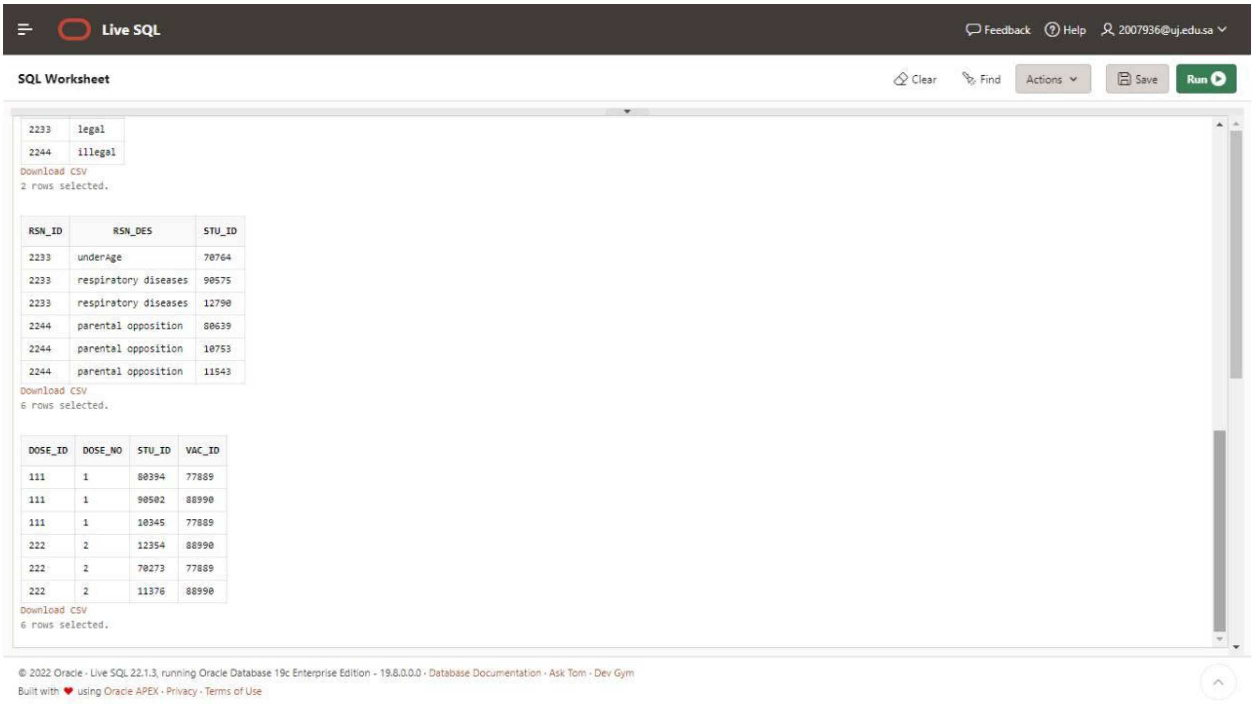
S_ID	DESCRIPTION	DOSE_ID
674	fever	111
765	headache	222

```
Download CSV
2 rows selected.
```

RSN_ID	RSN_TYP
2233	legal
2244	illegal

```
Download CSV
2 rows selected.
```

© 2022 Oracle · Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use



2233	legal
2244	illegal

Download CSV
2 rows selected.

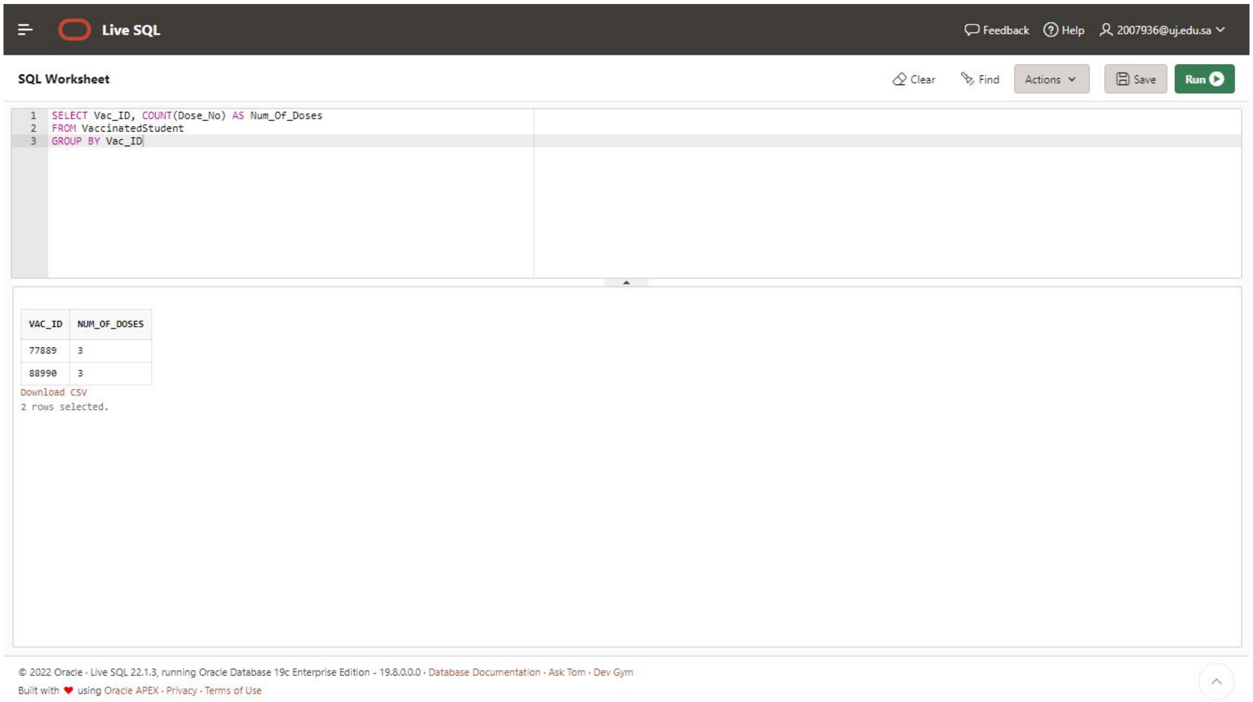
RSN_ID	RSN_DES	STU_ID
2233	underAge	70764
2233	respiratory diseases	90575
2233	respiratory diseases	12790
2244	parental opposition	80639
2244	parental opposition	10753
2244	parental opposition	11543

Download CSV
6 rows selected.

DOSE_ID	DOSE_NO	STU_ID	VAC_ID
111	1	80394	77889
111	1	90502	88990
111	1	10345	77889
222	2	12354	88990
222	2	70273	77889
222	2	11376	88990

Download CSV
6 rows selected.

© 2022 Oracle · Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ❤ using Oracle APEX · Privacy · Terms of Use



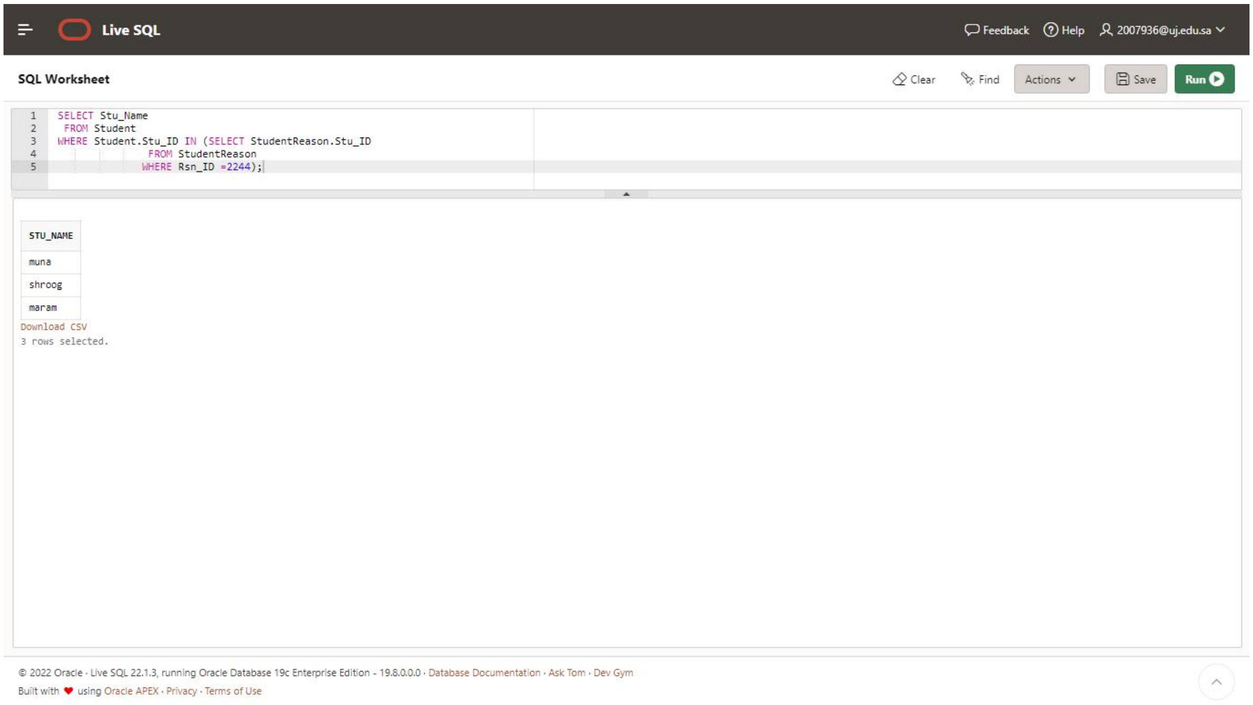
```
SELECT Vac_ID, COUNT(Dose_No) AS Num_Of_Doses
FROM VaccinatedStudent
GROUP BY Vac_ID
```

VAC_ID	NUM_OF_DOSES
77889	3
88990	3

Download CSV
2 rows selected.

© 2022 Oracle · Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ❤ using Oracle APEX · Privacy · Terms of Use

-This query display how many doses student took based on the vaccine ID which represent the vaccine type, phfizer or moderna in a new column name.

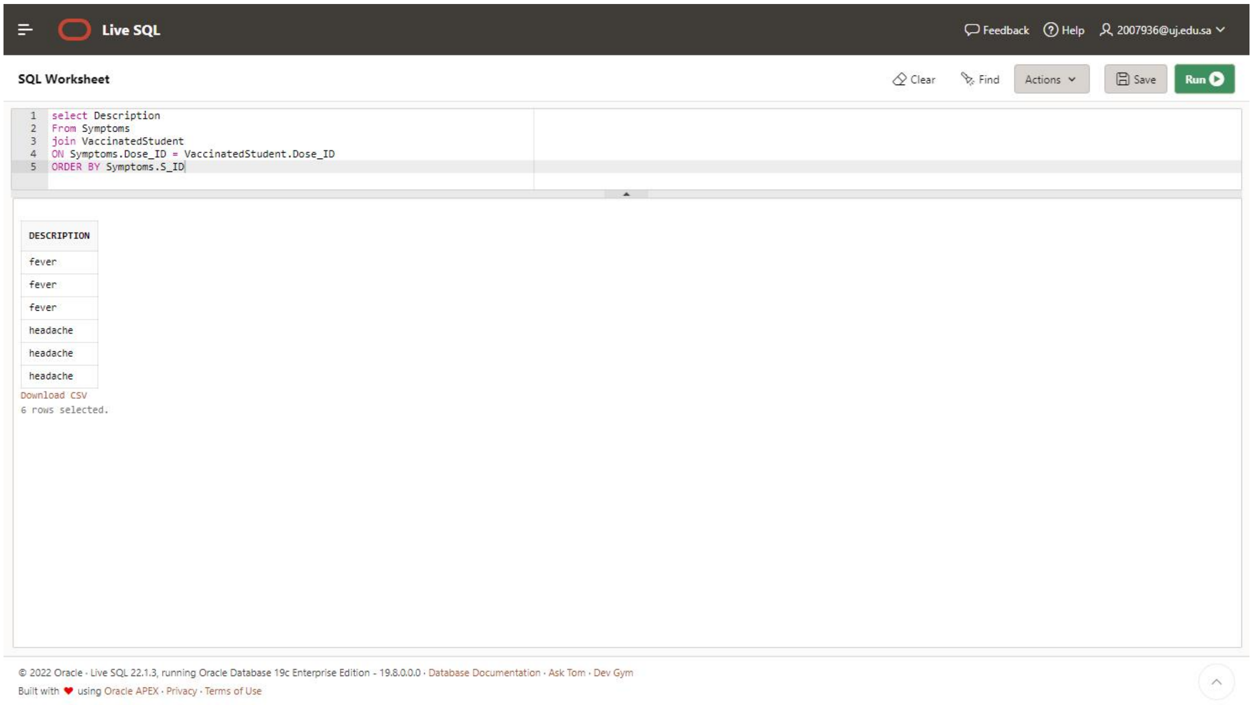


Live SQL
Feedback
Help
2007936@uj.edu.sa
SQL Worksheet
Clear
Find
Actions
Save
Run
© 2022 Oracle · Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

```sql
SELECT Stu_Name
 FROM Student
WHERE Student.Stu_ID IN (SELECT StudentReason.Stu_ID
                          FROM StudentReason
                         WHERE Rsn_ID =2244);
```

```
STU_NAME
muna
shroog
maram
Download CSV
3 rows selected.
```

-This sub-query shows the names of the students who have a legal reason for not completing the doses or for not taking them in the first place.

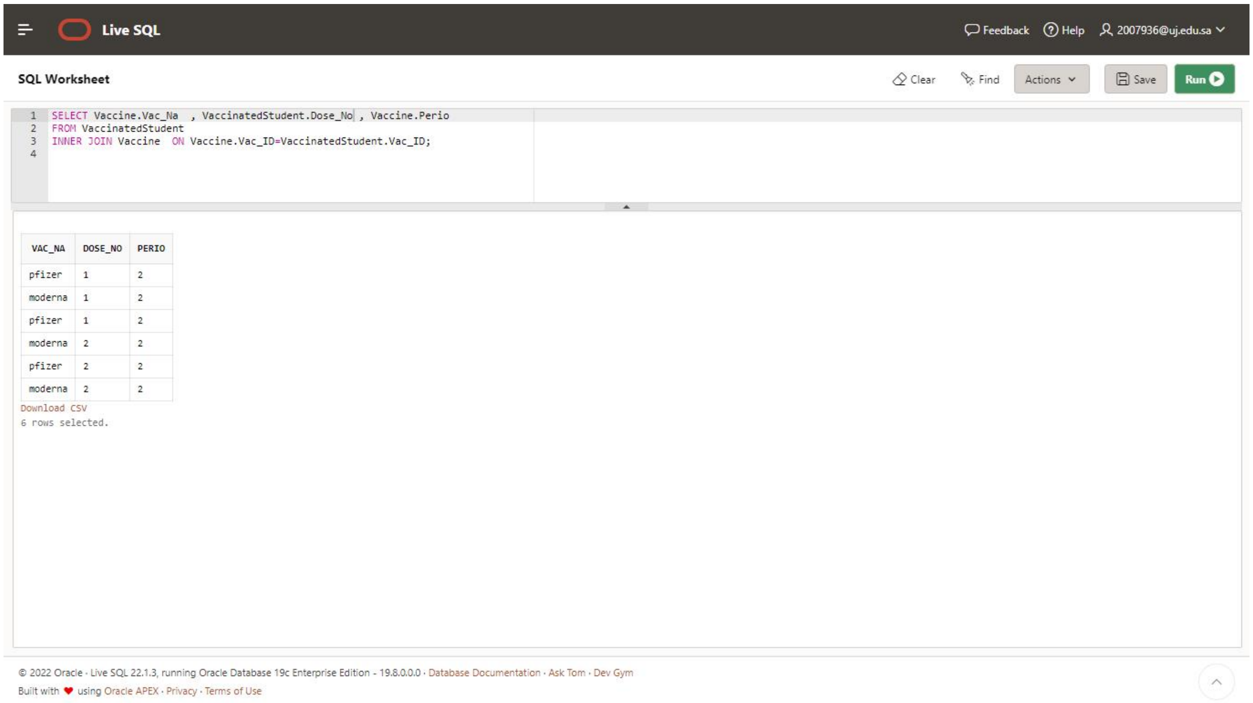


Live SQL
Feedback
Help
2007936@uj.edu.sa
SQL Worksheet
Clear
Find
Actions
Save
Run
© 2022 Oracle · Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

```sql
select Description
From Symptoms
join VaccinatedStudent
ON Symptoms.Dose_ID = VaccinatedStudent.Dose_ID
ORDER BY Symptoms.S_ID
```

```
DESCRIPTION
fever
fever
fever
headache
headache
headache
Download CSV
6 rows selected.
```

-This join query represents the symptoms of the vaccine based on the Id number of the dose because it might change either with it or with the type of the vaccine , also it's presented in order of symptoms id.



```sql
SELECT Vaccine.Vac_Na  , VaccinatedStudent.Dose_No , Vaccine.Perio
FROM VaccinatedStudent
INNER JOIN Vaccine  ON Vaccine.Vac_ID=VaccinatedStudent.Vac_ID;
```

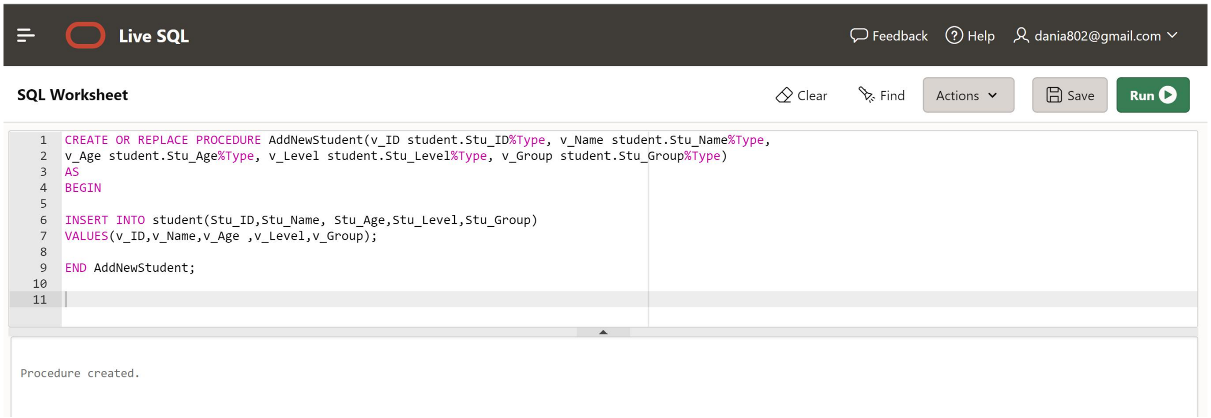
VAC_NA	DOSE_NO	PERIO
pfizer	1	2
moderna	1	2
pfizer	1	2
moderna	2	2
pfizer	2	2
moderna	2	2

```
Download CSV
6 rows selected.
```

© 2022 Oracle · Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ❤ using Oracle APEX · Privacy · Terms of Use

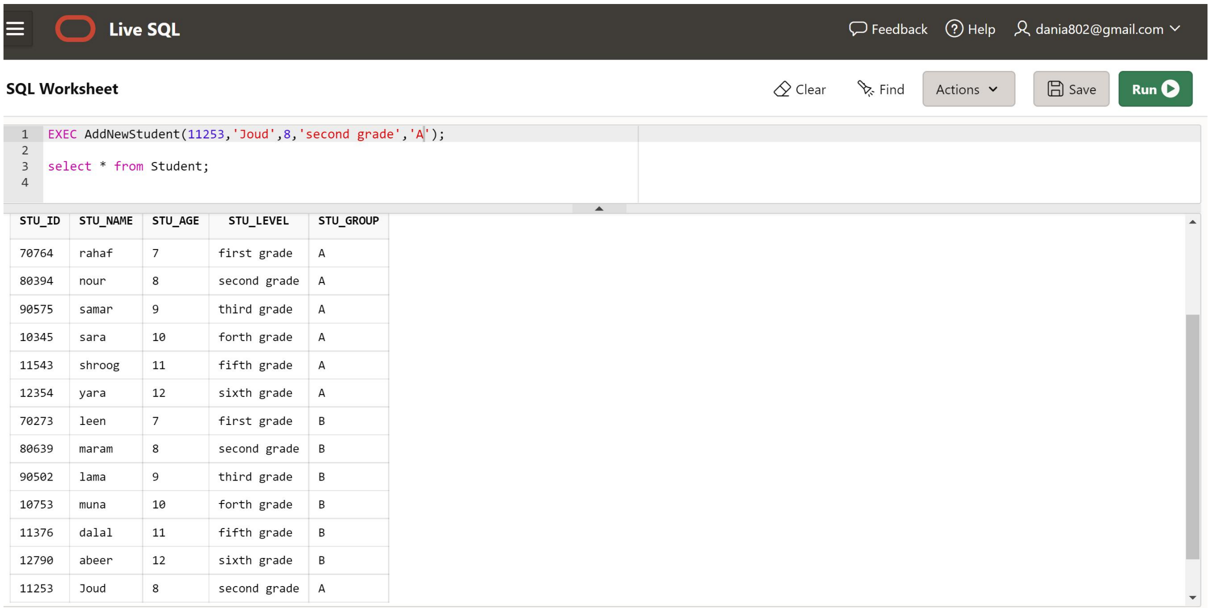
-This join query print the vaccine name with the dose number and the period that was between the doses.

This procedure to Add new student to the record list



```sql
CREATE OR REPLACE PROCEDURE AddNewStudent(v_ID student.Stu_ID%Type, v_Name student.Stu_Name%Type,
v_Age student.Stu_Age%Type, v_Level student.Stu_Level%Type, v_Group student.Stu_Group%Type)
AS
BEGIN

INSERT INTO student(Stu_ID,Stu_Name, Stu_Age,Stu_Level,Stu_Group)
VALUES(v_ID,v_Name,v_Age ,v_Level,v_Group);

END AddNewStudent;

```

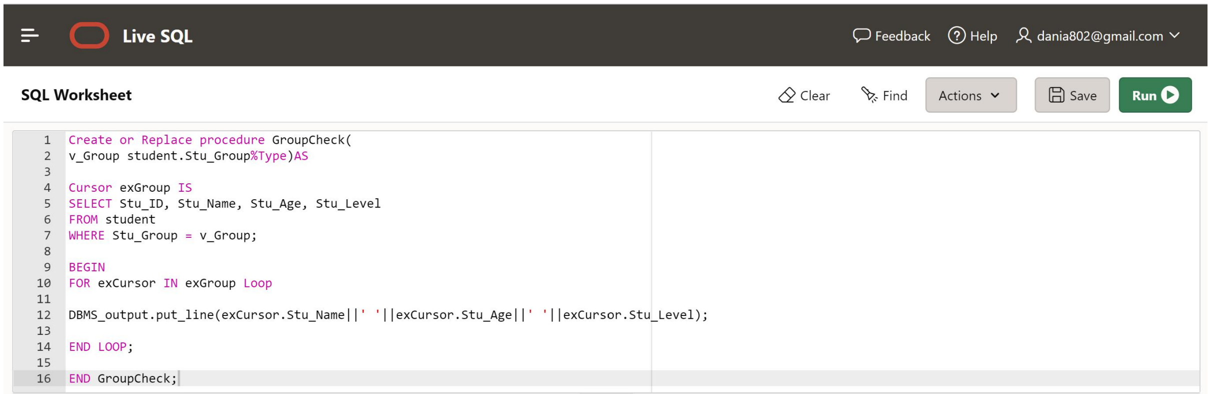
```
Procedure created.
```



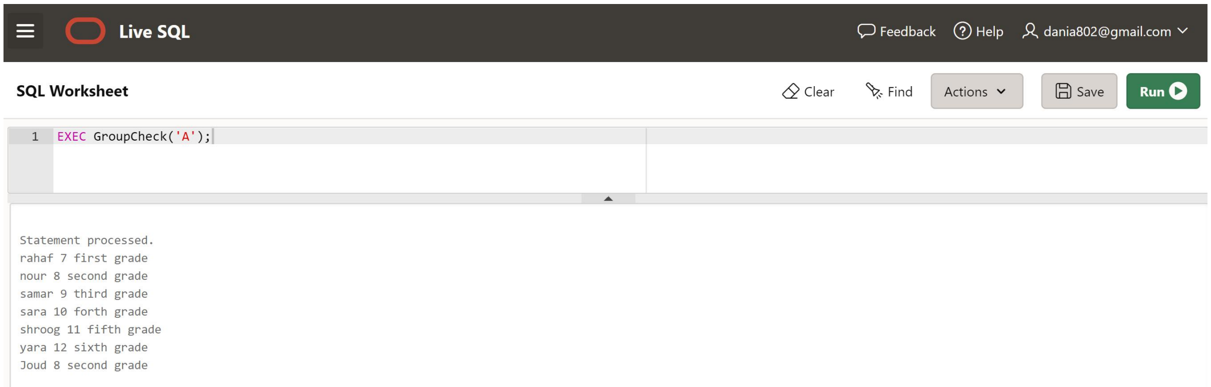
```
EXEC AddNewStudent(11253,'Joud',8,'second grade','A');

select * from Student;

```

STU_ID	STU_NAME	STU_AGE	STU_LEVEL	STU_GROUP
70764	rahaf	7	first grade	A
80394	nour	8	second grade	A
90575	samar	9	third grade	A
10345	sara	10	forth grade	A
11543	shroog	11	fifth grade	A
12354	yara	12	sixth grade	A
70273	leen	7	first grade	B
80639	maram	8	second grade	B
90502	lama	9	third grade	B
10753	muna	10	forth grade	B
11376	dalal	11	fifth grade	B
12790	abeer	12	sixth grade	B
11253	Joud	8	second grade	A

This procedure to return the list of students based on the group A or B



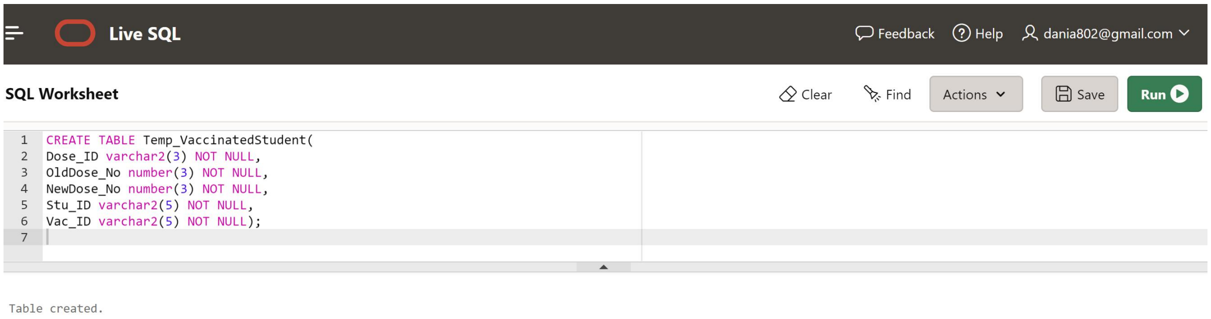
```
Create or Replace procedure GroupCheck(
v_Group student.Stu_Group%Type)AS

Cursor exGroup IS
SELECT Stu_ID, Stu_Name, Stu_Age, Stu_Level
FROM student
WHERE Stu_Group = v_Group;

BEGIN
FOR exCursor IN exGroup Loop

DBMS_output.put_line(exCursor.Stu_Name||' '||exCursor.Stu_Age||' '||exCursor.Stu_Level);

END LOOP;

END GroupCheck;
```



```
EXEC GroupCheck('A');
```

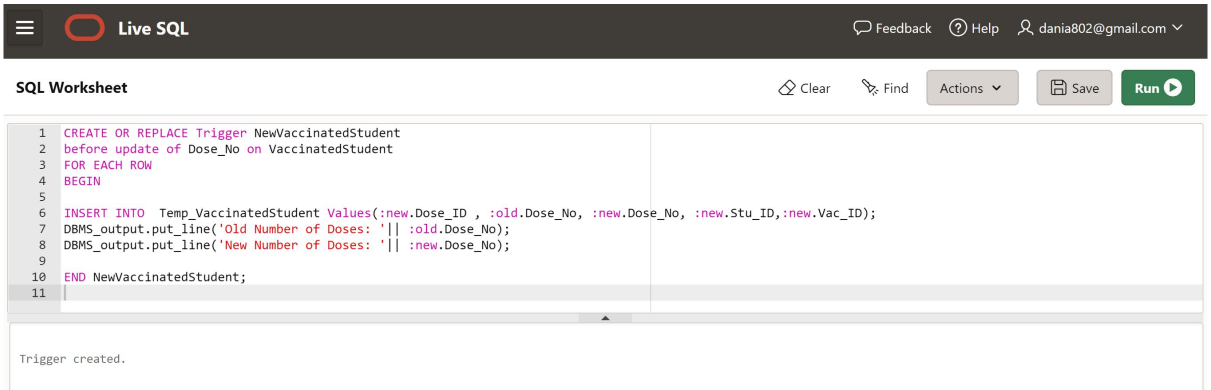
```
Statement processed.
rahaf 7 first grade
nour 8 second grade
samar 9 third grade
sara 10 forth grade
shroog 11 fifth grade
yara 12 sixth grade
Joud 8 second grade
```

-This is Trigger to update the number of doses for grade six student.



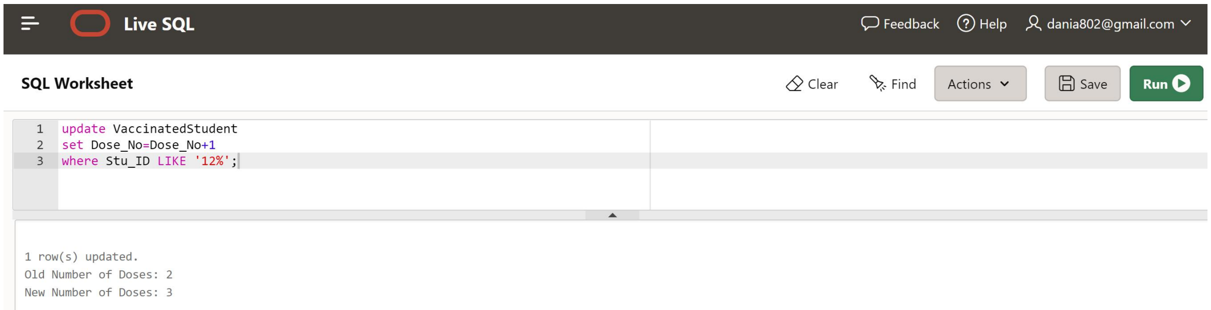
```
CREATE TABLE Temp_VaccinatedStudent(
Dose_ID varchar2(3) NOT NULL,
OldDose_No number(3) NOT NULL,
NewDose_No number(3) NOT NULL,
Stu_ID varchar2(5) NOT NULL,
Vac_ID varchar2(5) NOT NULL);
```

```
Table created.
```



```
CREATE OR REPLACE Trigger NewVaccinatedStudent
before update of Dose_No on VaccinatedStudent
FOR EACH ROW
BEGIN

INSERT INTO  Temp_VaccinatedStudent Values(:new.Dose_ID , :old.Dose_No, :new.Dose_No, :new.Stu_ID,:new.Vac_ID);
DBMS_output.put_line('Old Number of Doses: '|| :old.Dose_No);
DBMS_output.put_line('New Number of Doses: '|| :new.Dose_No);

END NewVaccinatedStudent;
```

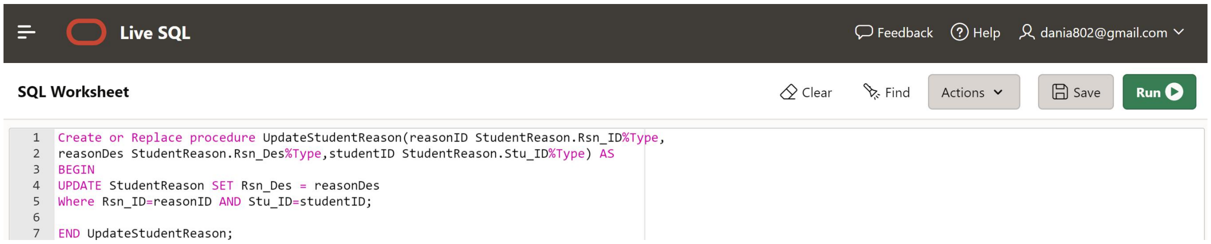
```
Trigger created.
```



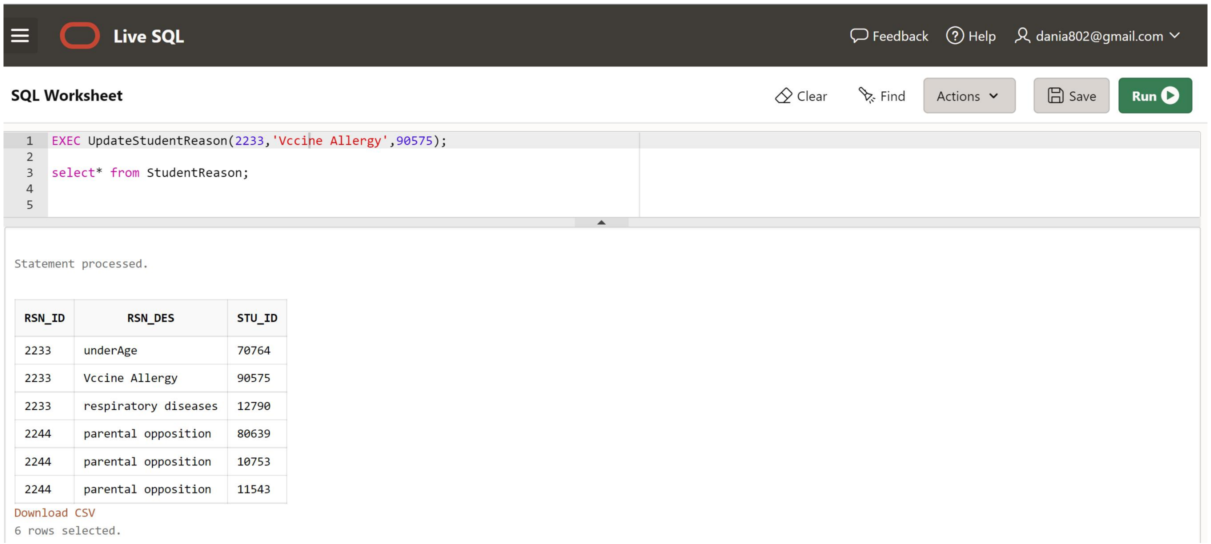
```
update VaccinatedStudent
set Dose_No=Dose_No+1
where Stu_ID LIKE '12%';
```

```
1 row(s) updated.
Old Number of Doses: 2
New Number of Doses: 3
```

This Procedure to Change the Reason for not Taking the vaccine



```
Create or Replace procedure UpdateStudentReason(reasonID StudentReason.Rsn_ID%Type,
reasonDes StudentReason.Rsn_Des%Type,studentID StudentReason.Stu_ID%Type) AS
BEGIN
UPDATE StudentReason SET Rsn_Des = reasonDes
Where Rsn_ID=reasonID AND Stu_ID=studentID;

END UpdateStudentReason;
```



```
EXEC UpdateStudentReason(2233,'Vccine Allergy',90575);

select* from StudentReason;
```

```
Statement processed.
```

RSN_ID	RSN_DES	STU_ID
2233	underAge	70764
2233	Vccine Allergy	90575
2233	respiratory diseases	12790
2244	parental opposition	80639
2244	parental opposition	10753
2244	parental opposition	11543

Download CSV

```
6 rows selected.
```