Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

ONLINE RESORT RESERVATION SYSTEM

A Project Report

for

DATABASE MANAGEMENT SYSTEMS

In

B.Tech -Information Technology and Engineering

By

EZHILARASI K (19BIT0395)

SARANYA M (20BIT0303)

ABRETHA BEGAM A (20BIT0309)

SLOT-G1+TG1

Under the Guidance of

PARIMALA M

Associate Professor Grade 1

Introduction

Basically the only use of DBMS is as the name says to manage a database system. Wherever and in whatever application you need to manage a database, you uses a DBMS .

1.Problem Statement

USER REQUIREMENTS:

Definition: To make the system fast and flexible, to reduce errors & expenses

and save time.

- ♣ Easy to handle
- ♣ Easy to update data
- ♣ Easy to record data
- ♣ We can also keep a backfile if the loaded files are crashed or erased.

FUNCTIONAL SYSTEM REQUIREMENTS

For online resort reservations, you need to have an Online resort Reservation System, as it provides a platform for both resort managers and customers. Resort managers can manage their resort rooms and customers can book rooms through the website generated by this system.

1. Launch a Resort Website -

You can look for a resort reservation system that can help you launch your own resort website.In this way, you'll be saving a lot of your money and save time that would be consumed to build one.

2. User-Friendly Interface

A resort booking engine with a good user-friendly interface will allow customers to easily navigate through your website. Thus, increasing the chances for booking as customers prefer to stay longer on a seamless website.

3. Easy Booking Process

The booking process should include minimal steps. If it takes a long time to fill unnecessary forms, chances are that the customer will abandon the booking and head to other websites.

4. Offline Booking

A resort booking should be able to offer you options to handle your on-desk and online bookings. If a customer walks in you should be able to book rooms from the back-end of the system. In addition to that, you must be able to keep an eye on the online bookings made through your websites.

5. Mobile-Friendly

As most of the customers nowadays use their phones, tablets, iPad etc ..devices to book rooms, it is an impossible to avoid feature.

6. Support for Multiple Languages & Currencies

As you launch a website, it is available on the internet. Now the customers around the world will be able to access your website. But what if they want to come to your resort and aren't able to understand or read your website in their language. If they don't understand then they will go for another resort website that helps them to learn more about their services. So make sure that the resort booking engine you choose has support for multiple Languages and currencies.

7. Integrated Payment Gateway

The booking will be done online so the customers will pay through their debit cards or credit cards. As security for online payments a high priority, it is really important to go for a resort booking engine that provides a secure payment gateway so that customers trust your brand and don't change their minds.

8. Easy Search Option

Customers must able search for resorts in a particular location and view the availability of rooms in between the preferred dates.

9. Discounts

Offering discounts is a great way to attract customers. Through the system, you should be ablet o create offers or discounts for new or current customers when required

2.Analysis

Entities

Strong:

1) Room

In this entity we can find all the details of the room like room no.

Whether the room is single, double or large bed. It also has the rate of

each room per day.

2) Guest

In this entity we will have the details of the guest as in their id, name,

phone no and address .

3) Bills

This entity has every bill no, amount and date of payment.

4) Food

This entity is exclusively used to keep track of food ordered in each room.

It has the food id, rate and type.two Types are there veg and non veg.

5) Facilities

This entity record every other facility that has been used by guests. It

holds Facility id, rate and type.

6) Resort

This entity has resort name, id, phone no and address.

7)Workers

This entity has id,name,address and workers phone no.

Different types of attributes:

Composite Attributes:

Type attribute of room entity consists single,double and large .type attribute of food entity ,it consists veg and non veg attributes.

Single Attributes:

Room entity has room no,rate single attributes.resort has name,id, address attributes.worker has the id,name,add single attributes.food has id,rate single attributes.facility entity has id,type rate single attributes.bills have date ,amount ,bill no single attributes.guest has add,name,id 3 single attributes.

Multivalue Attribute:

Phone number of resort,guest,worker entity is the multivalued attribute.

Descriptive Attribute:

Books is a relation between room and guest entites.it has 2 attributes check of date and check in date.orders is a relation between guest and food entites.it has 3 descriptive attributes quantity,time,date. Pays is the relation between guest and bills entites.here 1 descriptive attribute that is paying method.

Relationships and cardinality:

Guest Stays In A Room (1-1):

At a time one guest can stay in one room and room can occupy only one guest. So the relationship is (1-1).

Guest Paying Bills (1-1)

Guest while leaving the resort pays the for the whole accommodation in one bill with a unique bill id. Each bill has a unique bill id so one bill ca only belong to one customer. So the relationship is (1-1). Each bill is associated to a particular guest and each guest is associated to a bill id.

Guest Orders Food (N-N):

Each guest can order food from Room service. The options given to guest in the menu are plenty so they have a lot of choice to choose from. And one type of food can be ordered by many guests. So the relationship is (N-N). Guests need not necessarily order food similarly a particular food item need not be ordered by any guests so partial participation.

Guest Using Facilities (N-N):

Guests can use the facilities provided by the resort. This facilities include spa, gym, hot water spring, etc. They can choose any. And a specific facility can be availed by more than one guest at a time. So the relationship is (N-N).

Rooms In Resort (1-N):

Each resort must have more than one room and each room can only be there in one resort. So the relationship is (1-N). Each resort must have room and each room must be there in a resort so total participation from both entities.

Resort Manages Workers(1-N):

Workers are managed by a resort. Resort has many workers so here is 1:N relationship.

Participation:

Guests need to stay in a room but a room need not necessarily occupy a guest.so total participation .

It is not necessary for guests to use facilities and similarly each facility need not be used by guests. So partial participation.

Resort must have rooms. Similary rooms are in a resort so here is total participation.

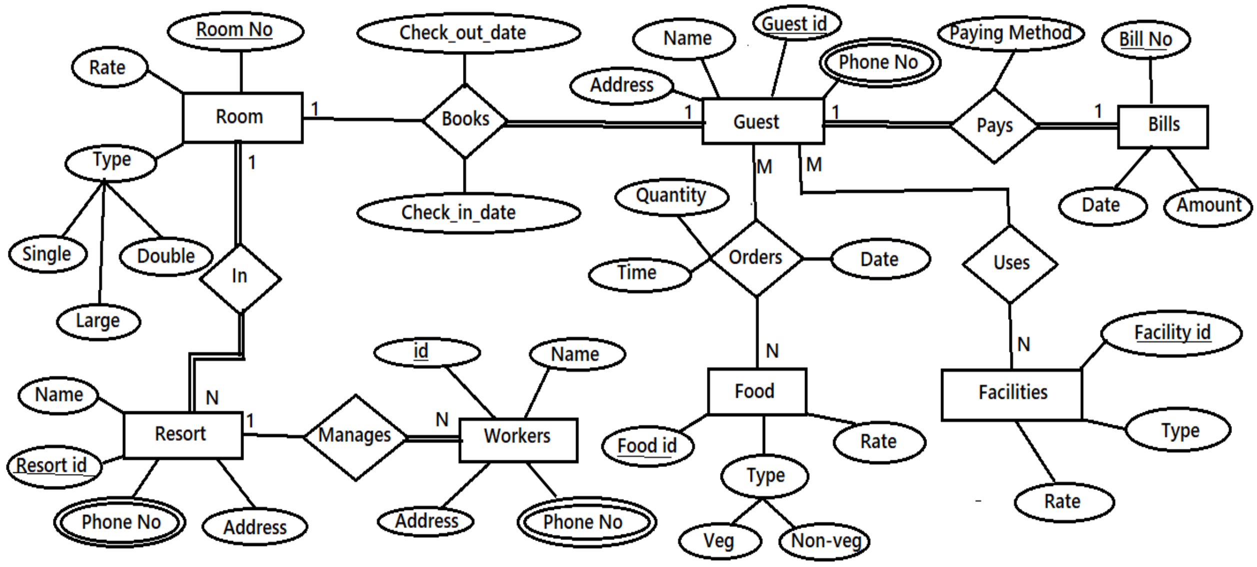
Guest pay bills that is mandatory similarly bills paid by a guest that is also compulsory.so total participation.

Guest orders food that foods are ordered by guest both are not compulsory.so partial participation.

Workers are compulsory managed by the resort.so total participation

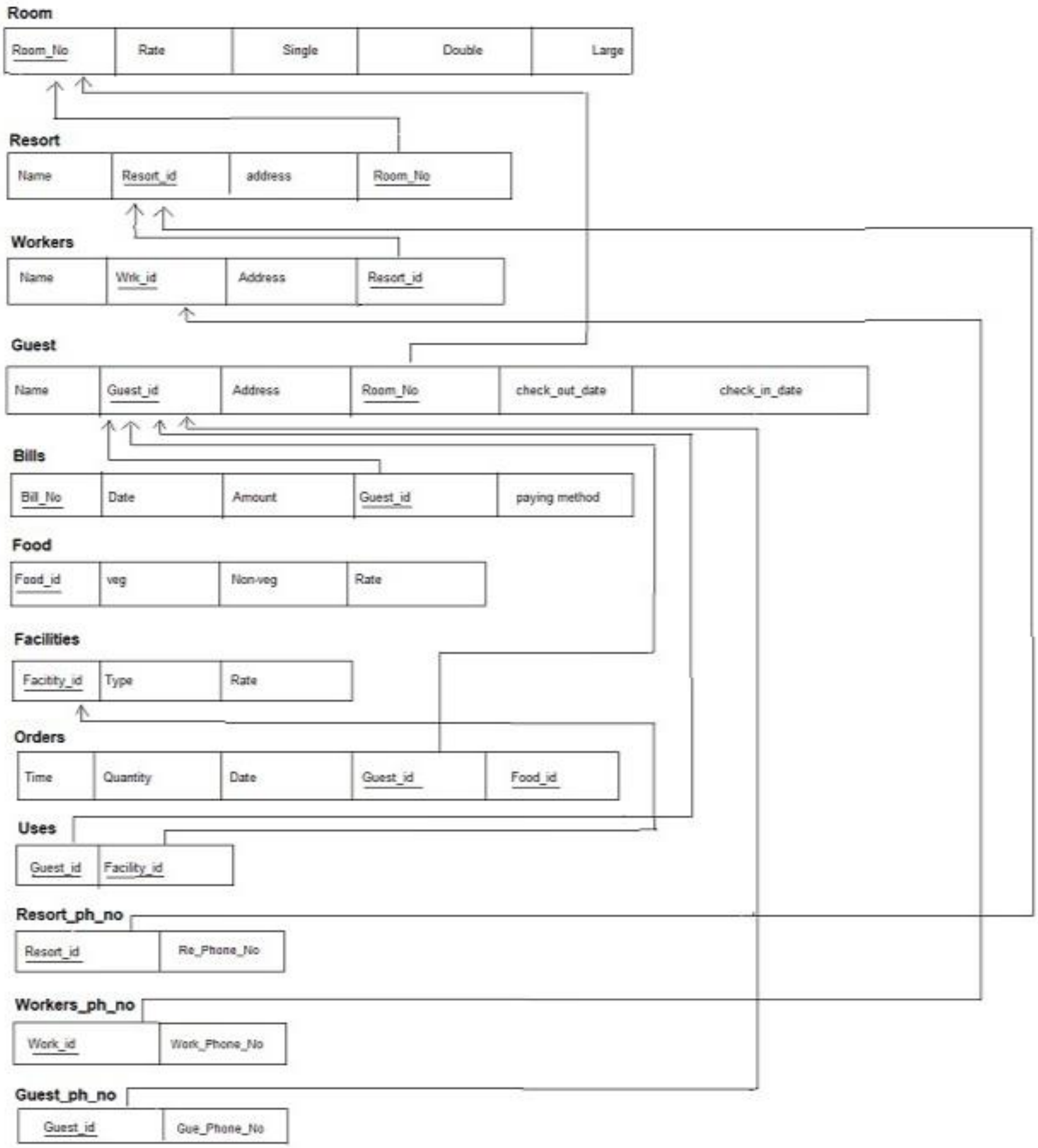
3.Design

ER Diagram :



4.Mapping

ER diagram to table using six guidelines .



Room

Room_No	Rate	Single	Double	Large

Resort

Name	Resort_id	address	Room_No

Workers

Name	Wrk_id	Address	Resort_id

Guest

Name	Guest_id	Address	Room_No	check_out_date	check_in_date

Bills

Bill_No	Date	Amount	Guest_id	paying method

Food

Food_id	veg	Non-veg	Rate

Facilities

Facitity_id	Type	Rate

Orders

Time	Quantity	Date	Guest_id	Food_id

Uses

Guest_id	Facility_id

Resort_ph_no

Resort_id	Re_Phone_No

Workers_ph_no

Work_id	Work_Phone_No

Guest_ph_no

Guest_id	Gue_Phone_No

5.Normalization

1NF

Definition:

1. It should only have single(atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

All the above tables are in 1NF.

1. Tables only have single(atomic) valued attributes/columns.
2. Values stored in a column also the same domain
3. All the columns in a table have unique names.
4. And the order in which data is stored, is not matter.

3NF

Definition :

1. The table should be in the Second Normal Form.
2. There should be no Transitive Functional Dependency.

Room

Rate → Single

Here Non -key attribute determines Non-key attribute. So it is in TFD.

So the relation is not in 3NF.We need to decompose the relation.

Room

Room_No	Rate	Single	Double	Large

Rate	Single

R1

Room_No	Rate	Double	Large

R2

Food

Veg→ Rate

Here Non -key attribute determines Non-key attribute. So it is in TFD.

So the relation is not in 3NF.We need to decompose the relation.

Transitive Rule holds true on **Food relation.**

Food

Food_id	veg	Non-veg	Rate

Veg	Rate

F21

Food_id	veg	Non-veg

F21

Facilities

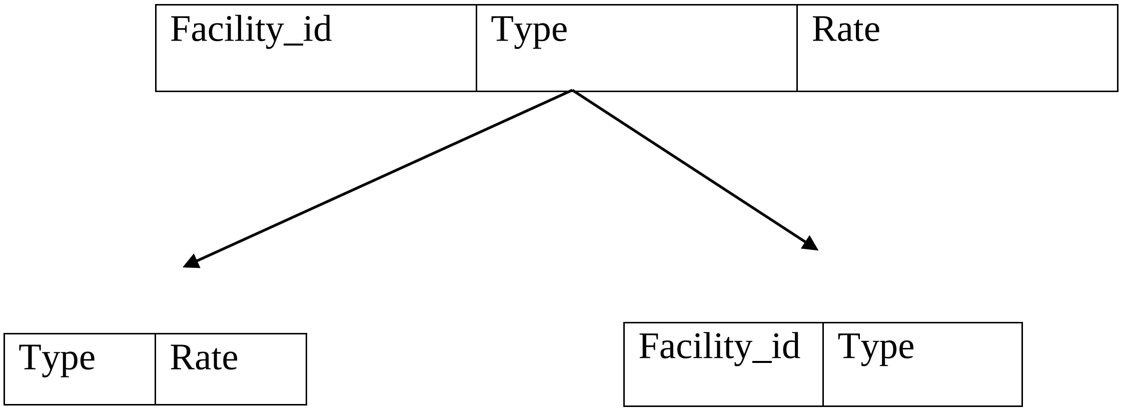
Type → Rate

Here Non -key attribute determines Non-key attribute. So it is in TFD.

So the relation is not in 3NF.We need to decompose the relation.

Transitive Rule holds true on **Facilities relation.**

Facilities



Facility_id	Type	Rate

Type	Rate

FA1

Facility_id	Type

FA2

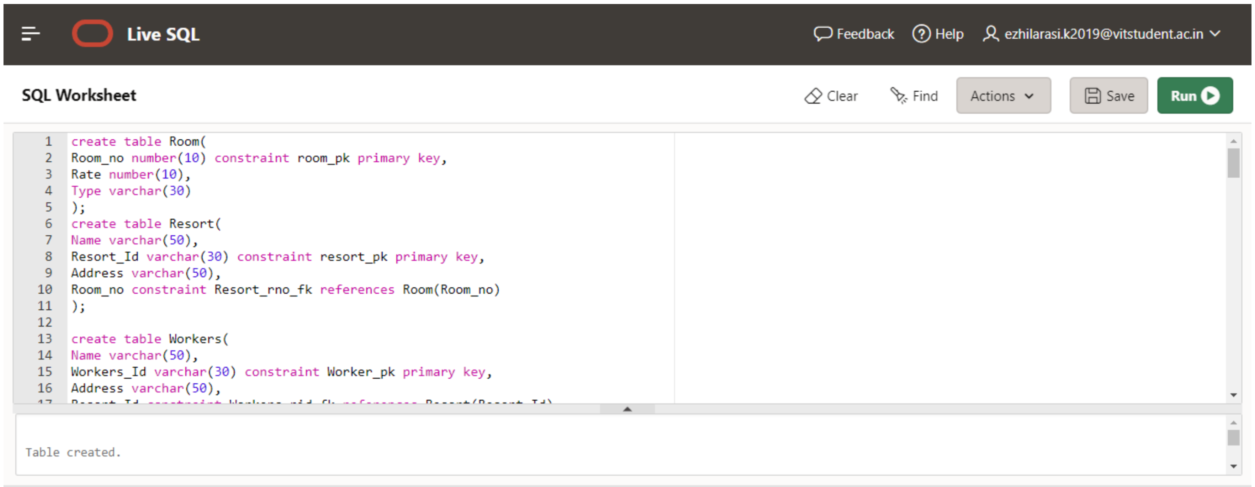
6.Implementation

CREATE TABLES:

1. **RESORT**
2. **ROOM**
3. **WORKERS**
4. **GUEST**
5. **BILLS**
6. **FACILITIES**
7. **FOOD**

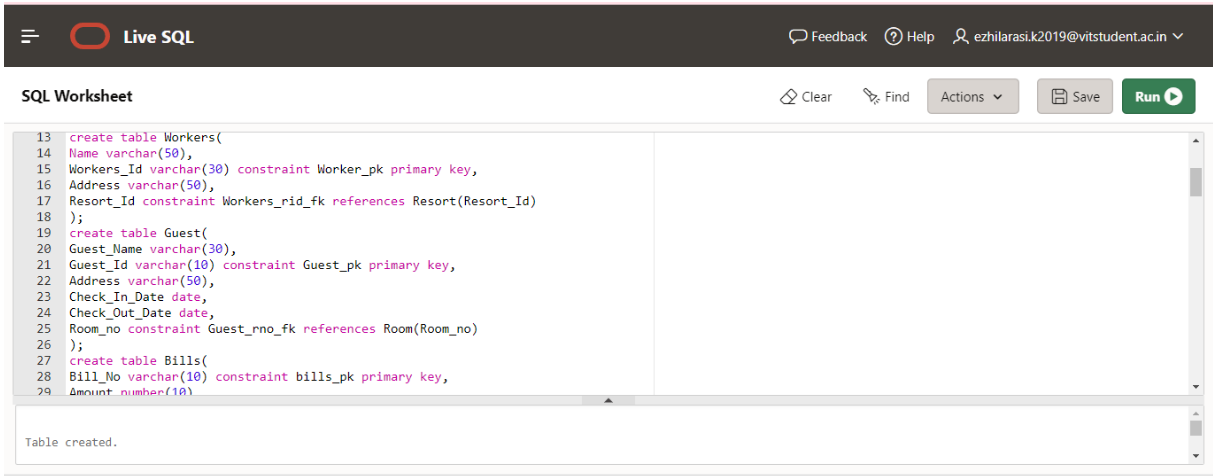
8. **ORDERS**
9. **GUEST_PHONE_NO**
10. **RESORT_PHONE_NO**
11. **WORKERS_PHONE_NO**
12. **USES**

```
create table Room(
Room_no number(10) constraint room_pk primary key,
Rate number(10),
Type varchar(30)
);
create table Resort(
Name varchar(50),
Resort_Id varchar(30) constraint resort_pk primary key,
Address varchar(50),
Room_no constraint Resort_rno_fk references Room(Room_no)
);

create table Workers(
Name varchar(50),
Workers_Id varchar(30) constraint Worker_pk primary key,
Address varchar(50),
Resort_Id constraint Workers_rid_fk references Resort(Resort_Id)
);
create table Guest(
Guest_Name varchar(30),
Guest_Id varchar(10) constraint Guest_pk primary key,
Address varchar(50),
Check_In_Date date,
Check_Out_Date date,
Room_no constraint Guest_rno_fk references Room(Room_no)
);
create table Bills(
Bill_No varchar(10) constraint bills_pk primary key,
Amount number(10),
Bill_Date date,
Paying_Method varchar(30),
Guest_Id constraint bills_Gid_fk references Guest(Guest_Id)
);
create table Food(
Food_Id varchar(30) constraint food_pk primary key,
Type varchar(30),
Rate number(10)
);

create table Facilities(
Facility_Id varchar(30) constraint facility_pk primary key,
Type varchar(30),
Rate number(10)
);
create table Orders(
Order_Date date,
Time Timestamp(0),
Quantity varchar(30),
Guest_Id constraint Orders_Gid_fk references Guest(Guest_Id),
Food_Id constraint Orders_fid_fk references Food(Food_Id)
);

create table Uses(
Guest_Id constraint Uses_Gid_fk references Guest(Guest_Id),
Facility_Id constraint Uses_fid_fk references Facilities(Facility_Id)
);

create table Resort_Phone_No(
Phone_no number(10),
Resort_Id constraint  Resortphno_Rid_fk  references Resort(Resort_Id)
);

create table Workers_Phone_No(
Phone_no number(10),
Workers_Id constraint Workersphno_wid_fk references Workers(Workers_Id)
);

create table Guest_Phone_No(
Phone_no number(10),
Guest_Id constraint Guestphno_Gid_fk references Guest(Guest_Id)
);
```



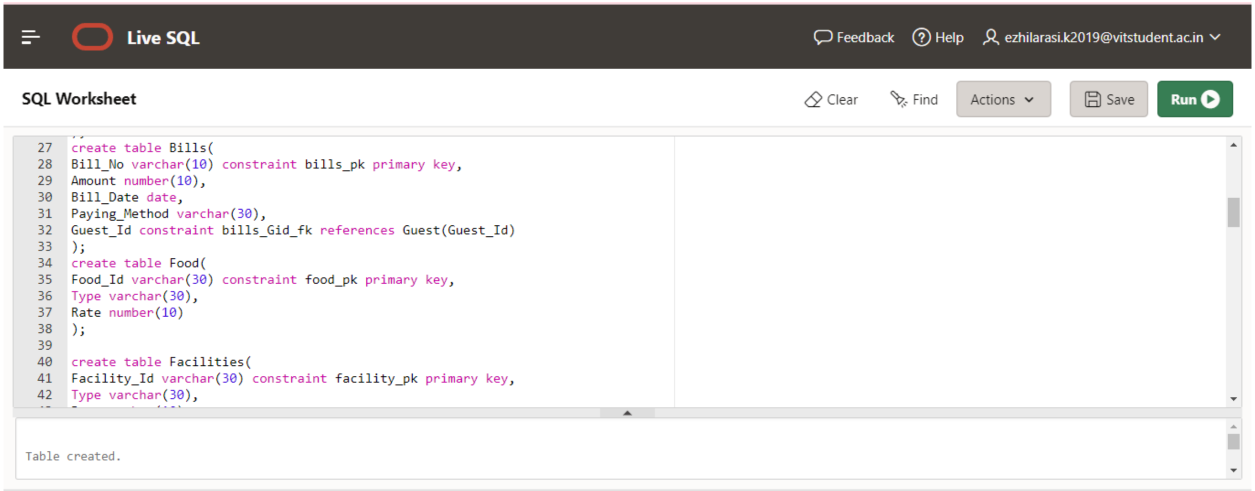
```
create table Room(
Room_no number(10) constraint room_pk primary key,
Rate number(10),
Type varchar(30)
);
create table Resort(
Name varchar(50),
Resort_Id varchar(30) constraint resort_pk primary key,
Address varchar(50),
Room_no constraint Resort_rno_fk references Room(Room_no)
);

create table Workers(
Name varchar(50),
Workers_Id varchar(30) constraint Worker_pk primary key,
Address varchar(50),
Resort_Id constraint Workers_rid_fk references Resort(Resort_Id)
```

```
Table created.
```



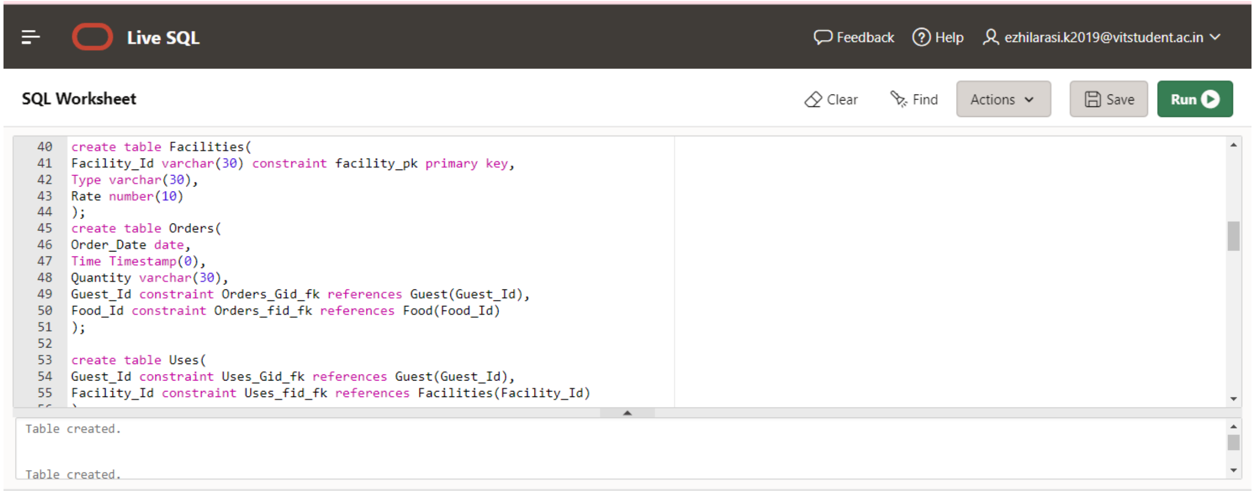
```
create table Workers(
Name varchar(50),
Workers_Id varchar(30) constraint Worker_pk primary key,
Address varchar(50),
Resort_Id constraint Workers_rid_fk references Resort(Resort_Id)
);
create table Guest(
Guest_Name varchar(30),
Guest_Id varchar(10) constraint Guest_pk primary key,
Address varchar(50),
Check_In_Date date,
Check_Out_Date date,
Room_no constraint Guest_rno_fk references Room(Room_no)
);
create table Bills(
Bill_No varchar(10) constraint bills_pk primary key,
Amount number(10)
```

```
Table created.
```



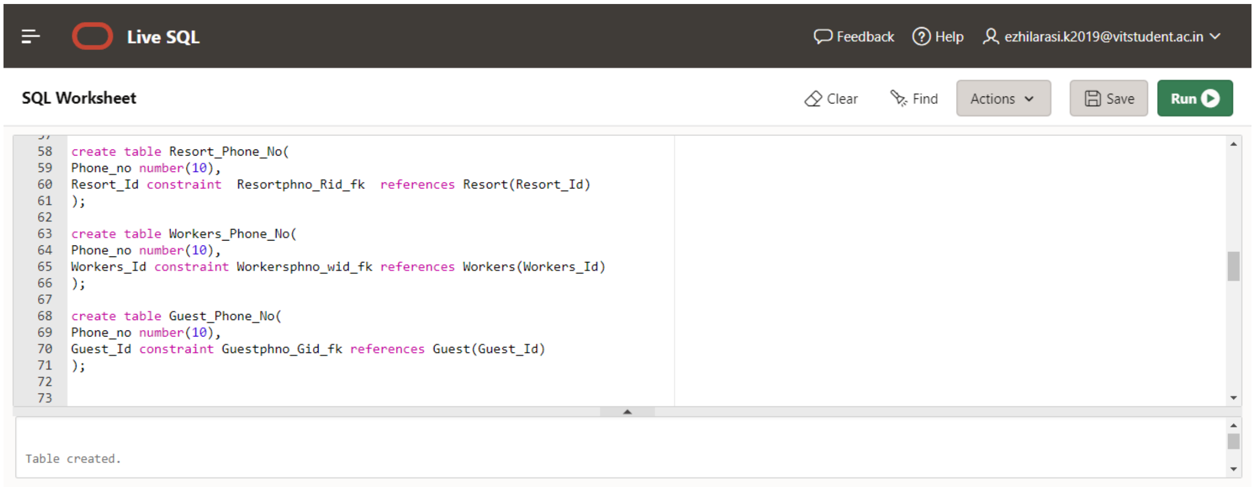
```
create table Bills(
Bill_No varchar(10) constraint bills_pk primary key,
Amount number(10),
Bill_Date date,
Paying_Method varchar(30),
Guest_Id constraint bills_Gid_fk references Guest(Guest_Id)
);
create table Food(
Food_Id varchar(30) constraint food_pk primary key,
Type varchar(30),
Rate number(10)
);

create table Facilities(
Facility_Id varchar(30) constraint facility_pk primary key,
Type varchar(30),
```

```
Table created.
```



```sql
create table Facilities(
Facility_Id varchar(30) constraint facility_pk primary key,
Type varchar(30),
Rate number(10)
);
create table Orders(
Order_Date date,
Time Timestamp(0),
Quantity varchar(30),
Guest_Id constraint Orders_Gid_fk references Guest(Guest_Id),
Food_Id constraint Orders_fid_fk references Food(Food_Id)
);

create table Uses(
Guest_Id constraint Uses_Gid_fk references Guest(Guest_Id),
Facility_Id constraint Uses_fid_fk references Facilities(Facility_Id)
```

```
Table created.

Table created.
```



```sql
create table Resort_Phone_No(
Phone_no number(10),
Resort_Id constraint  Resortphno_Rid_fk  references Resort(Resort_Id)
);

create table Workers_Phone_No(
Phone_no number(10),
Workers_Id constraint Workersphno_wid_fk references Workers(Workers_Id)
);

create table Guest_Phone_No(
Phone_no number(10),
Guest_Id constraint Guestphno_Gid_fk references Guest(Guest_Id)
);
```

```
Table created.
```

Inserting values:

insert into Room values('101',5000,'single');

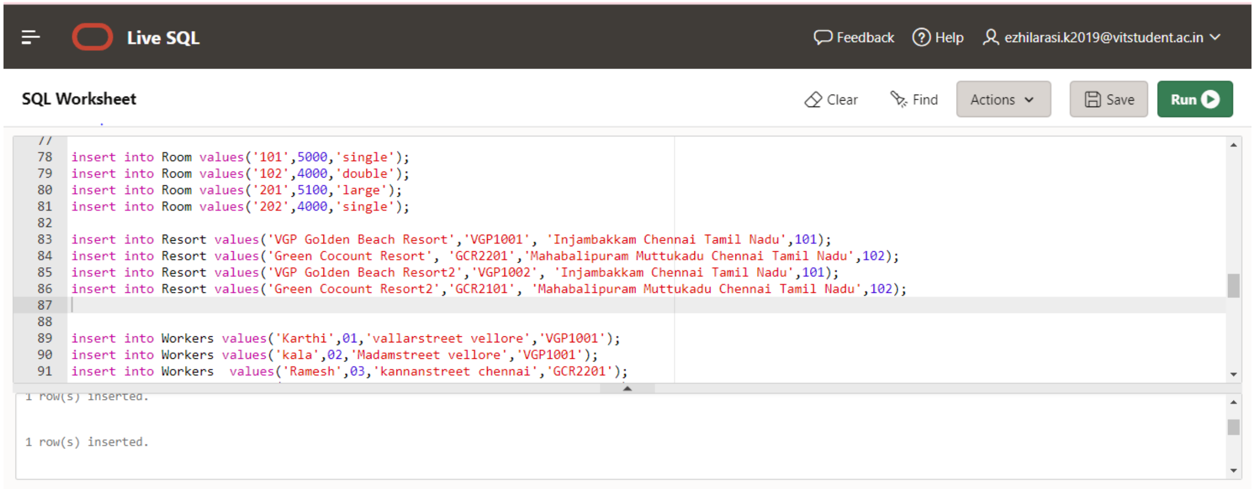
insert into Room values('102',4000,'double');
insert into Room values('201',5100,'large');
insert into Room values('202',4000,'single');

insert into Resort values('VGP Golden Beach Resort','VGP1001', 'Injambakkam Chennai Tamil Nadu',101);
insert into Resort values('Green Cocount Resort', 'GCR2201','Mahabalipuram Muttukadu Chennai Tamil Nadu',102);
insert into Resort values('VGP Golden Beach Resort2','VGP1002', 'Injambakkam Chennai Tamil Nadu',101);
insert into Resort values('Green Cocount Resort2','GCR2101', 'Mahabalipuram Muttukadu Chennai Tamil Nadu',102);

*

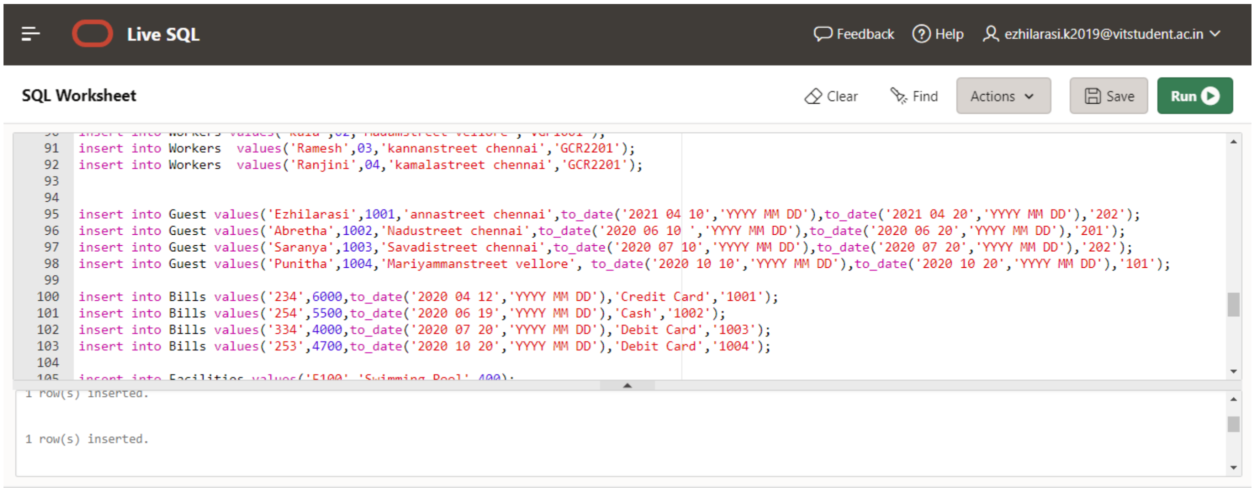
```
insert into Workers values('Karthi',01,'vallarstreet vellore','VGP1001');
insert into Workers values('kala',02,'Madamstreet vellore','VGP1001');
insert into Workers  values('Ramesh',03,'kannanstreet chennai','GCR2201');
insert into Workers  values('Ranjini',04,'kamalastreet chennai','GCR2201');


insert into Guest values('Ezhilarasi',1001,'annastreet chennai',to_date('2021 04
10','YYYY MM DD'),to_date('2021 04 20','YYYY MM DD'),'202');
insert into Guest values('Abretha',1002,'Nadustreet chennai',to_date('2020 06 10
','YYYY MM DD'),to_date('2020 06 20','YYYY MM DD'),'201');
insert into Guest values('Saranya',1003,'Savadistreet chennai',to_date('2020 07
10','YYYY MM DD'),to_date('2020 07 20','YYYY MM DD'),'202');
insert into Guest values('Punitha',1004,'Mariyammanstreet vellore', to_date('2020
10 10','YYYY MM DD'),to_date('2020 10 20','YYYY MM DD'),'101');

insert into Bills values('234',6000,to_date('2020 04 12','YYYY MM DD'),'Credit
Card','1001');
insert into Bills values('254',5500,to_date('2020 06 19','YYYY MM
DD'),'Cash','1002');
insert into Bills values('334',4000,to_date('2020 07 20','YYYY MM DD'),'Debit
Card','1003');
insert into Bills values('253',4700,to_date('2020 10 20','YYYY MM DD'),'Debit
Card','1004');

insert into Facilities values('F100','Swimming Pool',400);
insert into Facilities values('F110','outdoor restaurant',500);
insert into Facilities values('F200','outdoor restaurant',520);
insert into Facilities values('F210','Swimming Pool',600);

insert into Food values('V1','VEG',200);
insert into Food values('NV1','NON VEG',300);
insert into Food values('V2','VEG',250);
insert into Food values('NV2','NON VEG',200);

insert into Orders values(to_date('2020 04 15','YYYY MM
DD'),to_timestamp('8:00:00','HH24:MI:SS'),'1','1001','V1');
insert into Orders values(to_date('2020 06 10','YYYY MM
DD'),to_timestamp('9:00:00','HH24:MI:SS'),'1','1002','NV1');
```

```
insert into Orders values(to_date('2020 07 11','YYYY MM DD'),to_timestamp('8:30:00','HH24:MI:SS'),'1','1003','V2');
insert into Orders values(to_date('2020 10 18','YYYY MM DD'),to_timestamp('10:00:00','HH24:MI:SS'),'1','1004','NV2');

insert into Guest_Phone_No values(2345678192,'1001');
insert into Guest_Phone_No values(7745878192,'1002');
insert into Guest_Phone_No values(3342678192,'1003');
insert into Guest_Phone_No values(9048678195,'1004');


insert into Resort_Phone_No values(2178234677,'VGP1001');
insert into Resort_Phone_No values(9758364679,'VGP1001');
insert into Resort_Phone_No values(4178534672,'GCR2201');
insert into Resort_Phone_No values(9078634672,'GCR2201');

insert into Workers_Phone_No values(2170934677,01);
insert into Workers_Phone_No values(9753964679,02);
insert into Workers_Phone_No values(4176934672,03);
insert into Workers_Phone_No values(9048734672,04);
```



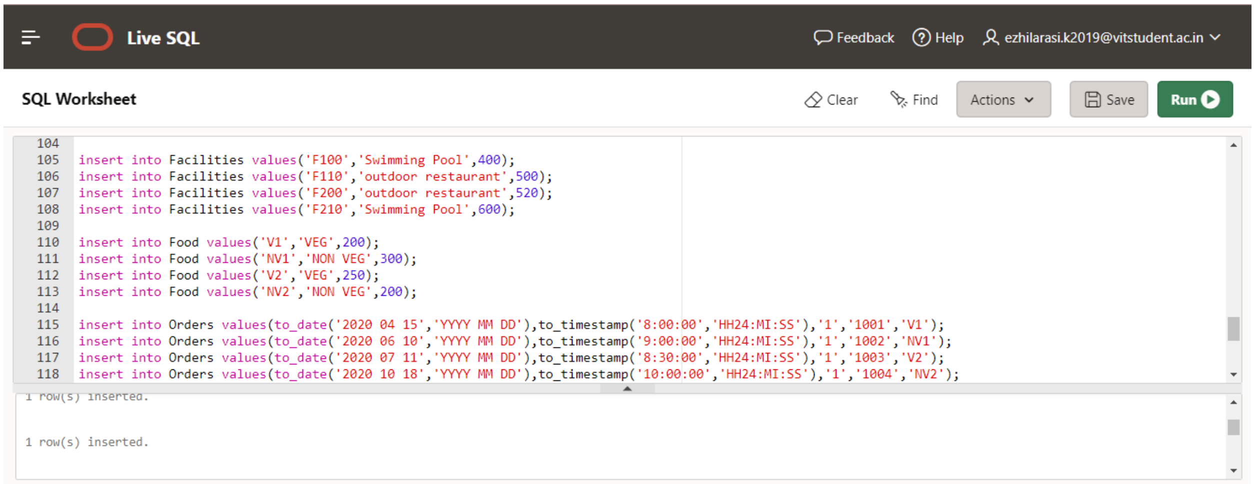
```
insert into Room values('101',5000,'single');
insert into Room values('102',4000,'double');
insert into Room values('201',5100,'large');
insert into Room values('202',4000,'single');

insert into Resort values('VGP Golden Beach Resort','VGP1001', 'Injambakkam Chennai Tamil Nadu',101);
insert into Resort values('Green Cocount Resort', 'GCR2201','Mahabalipuram Muttukadu Chennai Tamil Nadu',102);
insert into Resort values('VGP Golden Beach Resort2','VGP1002', 'Injambakkam Chennai Tamil Nadu',101);
insert into Resort values('Green Cocount Resort2','GCR2101', 'Mahabalipuram Muttukadu Chennai Tamil Nadu',102);


insert into Workers values('Karthi',01,'vallarstreet vellore','VGP1001');
insert into Workers values('kala',02,'Madamstreet vellore','VGP1001');
insert into Workers  values('Ramesh',03,'kannanstreet chennai','GCR2201');
```

```
1 row(s) inserted.


1 row(s) inserted.
```



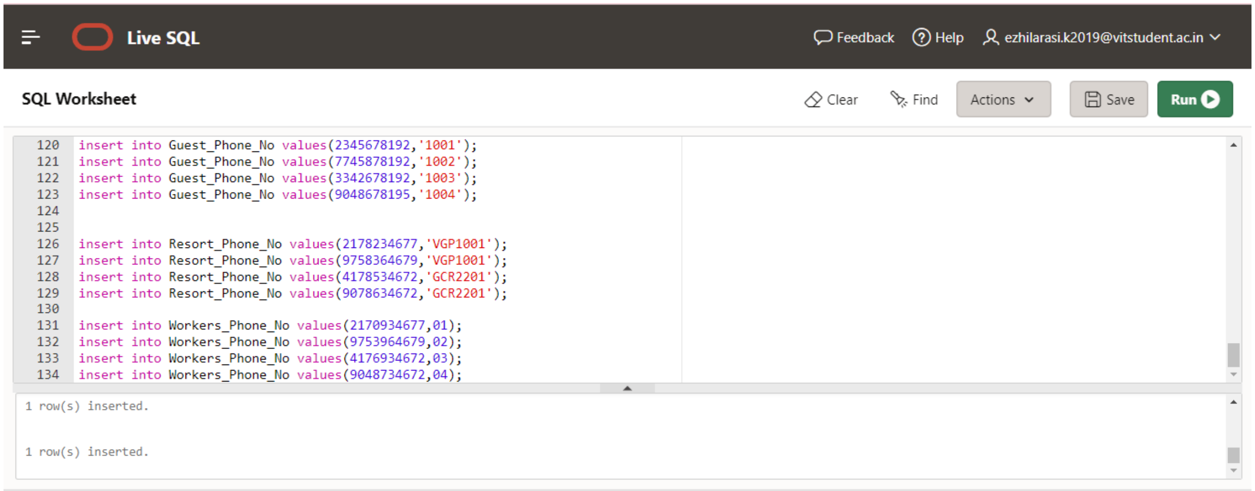
```
insert into Workers  values('Ramesh',03,'kannanstreet chennai','GCR2201');
insert into Workers  values('Ranjini',04,'kamalastreet chennai','GCR2201');


insert into Guest values('Ezhilarasi',1001,'annastreet chennai',to_date('2021 04 10','YYYY MM DD'),to_date('2021 04 20','YYYY MM DD'),'202');
insert into Guest values('Abretha',1002,'Nadustreet chennai',to_date('2020 06 10 ','YYYY MM DD'),to_date('2020 06 20','YYYY MM DD'),'201');
insert into Guest values('Saranya',1003,'Savadistreet chennai',to_date('2020 07 10','YYYY MM DD'),to_date('2020 07 20','YYYY MM DD'),'202');
insert into Guest values('Punitha',1004,'Mariyammanstreet vellore', to_date('2020 10 10','YYYY MM DD'),to_date('2020 10 20','YYYY MM DD'),'101');

insert into Bills values('234',6000,to_date('2020 04 12','YYYY MM DD'),'Credit Card','1001');
insert into Bills values('254',5500,to_date('2020 06 19','YYYY MM DD'),'Cash','1002');
insert into Bills values('334',4000,to_date('2020 07 20','YYYY MM DD'),'Debit Card','1003');
insert into Bills values('253',4700,to_date('2020 10 20','YYYY MM DD'),'Debit Card','1004');

insert into Facilities values('F100','Swimming Pool',400);
```

```
1 row(s) inserted.


1 row(s) inserted.
```



```

insert into Facilities values('F100','Swimming Pool',400);
insert into Facilities values('F110','outdoor restaurant',500);
insert into Facilities values('F200','outdoor restaurant',520);
insert into Facilities values('F210','Swimming Pool',600);

insert into Food values('V1','VEG',200);
insert into Food values('NV1','NON VEG',300);
insert into Food values('V2','VEG',250);
insert into Food values('NV2','NON VEG',200);

insert into Orders values(to_date('2020 04 15','YYYY MM DD'),to_timestamp('8:00:00','HH24:MI:SS'),'1','1001','V1');
insert into Orders values(to_date('2020 06 10','YYYY MM DD'),to_timestamp('9:00:00','HH24:MI:SS'),'1','1002','NV1');
insert into Orders values(to_date('2020 07 11','YYYY MM DD'),to_timestamp('8:30:00','HH24:MI:SS'),'1','1003','V2');
insert into Orders values(to_date('2020 10 18','YYYY MM DD'),to_timestamp('10:00:00','HH24:MI:SS'),'1','1004','NV2');
```

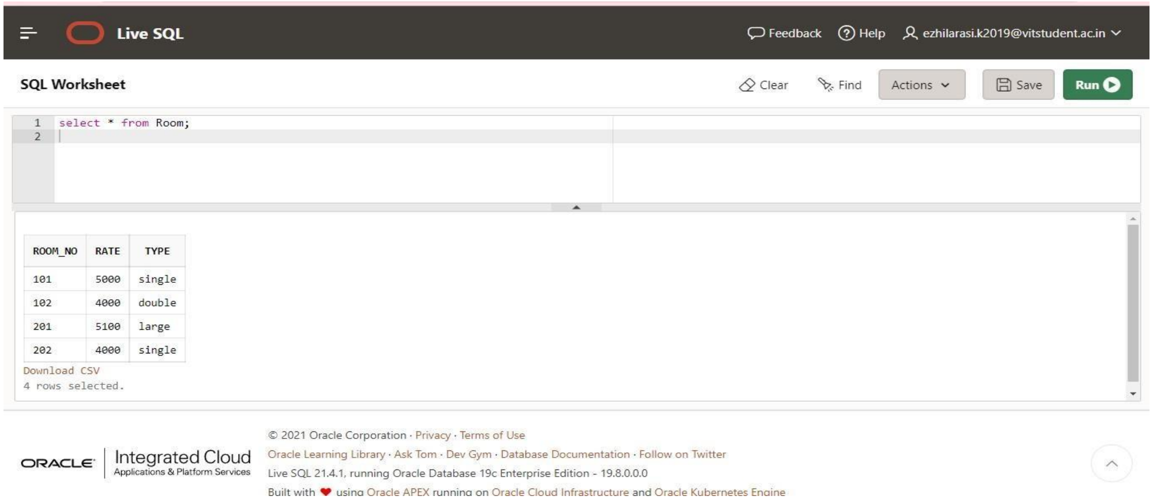
```
1 row(s) inserted.


1 row(s) inserted.
```



```
insert into Guest_Phone_No values(2345678192,'1001');
insert into Guest_Phone_No values(7745878192,'1002');
insert into Guest_Phone_No values(3342678192,'1003');
insert into Guest_Phone_No values(9048678195,'1004');


insert into Resort_Phone_No values(2178234677,'VGP1001');
insert into Resort_Phone_No values(9758364679,'VGP1001');
insert into Resort_Phone_No values(4178534672,'GCR2201');
insert into Resort_Phone_No values(9078634672,'GCR2201');

insert into Workers_Phone_No values(2170934677,01);
insert into Workers_Phone_No values(9753964679,02);
insert into Workers_Phone_No values(4176934672,03);
insert into Workers_Phone_No values(9048734672,04);
```

```
1 row(s) inserted.


1 row(s) inserted.
```

Tables

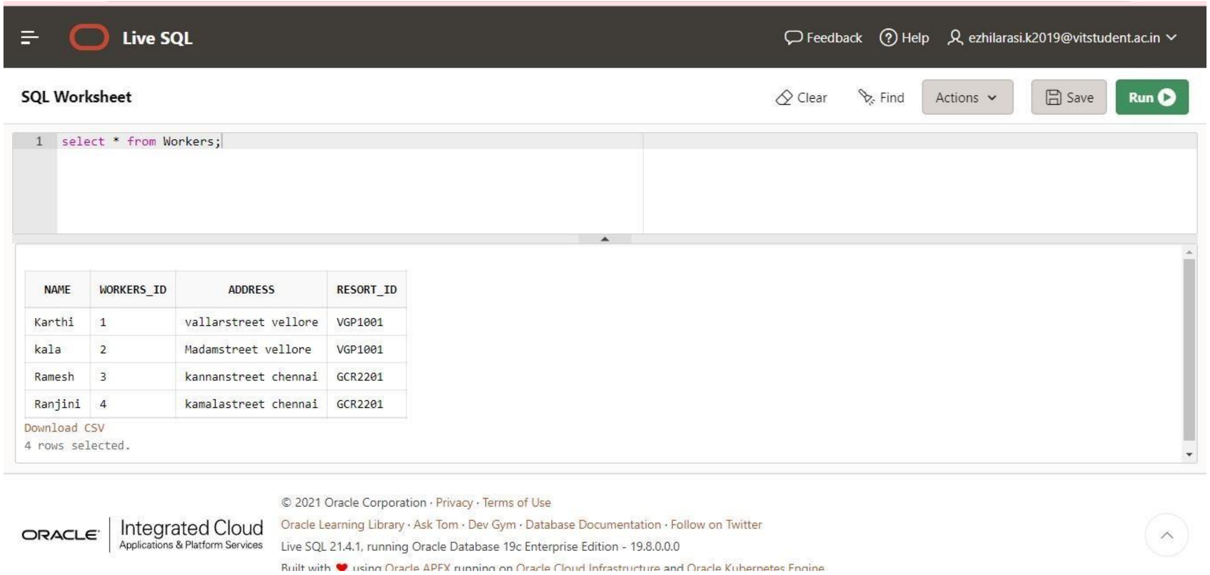


```sql
select * from Room;
```

ROOM_NO	RATE	TYPE
101	5000	single
102	4000	double
201	5100	large
202	4000	single

Download CSV
4 rows selected.

© 2021 Oracle Corporation · Privacy · Terms of Use
Oracle Learning Library · Ask Tom · Dev Gym · Database Documentation · Follow on Twitter
Live SQL 21.4.1, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0
Built with ❤ using Oracle APEX running on Oracle Cloud Infrastructure and Oracle Kubernetes Engine

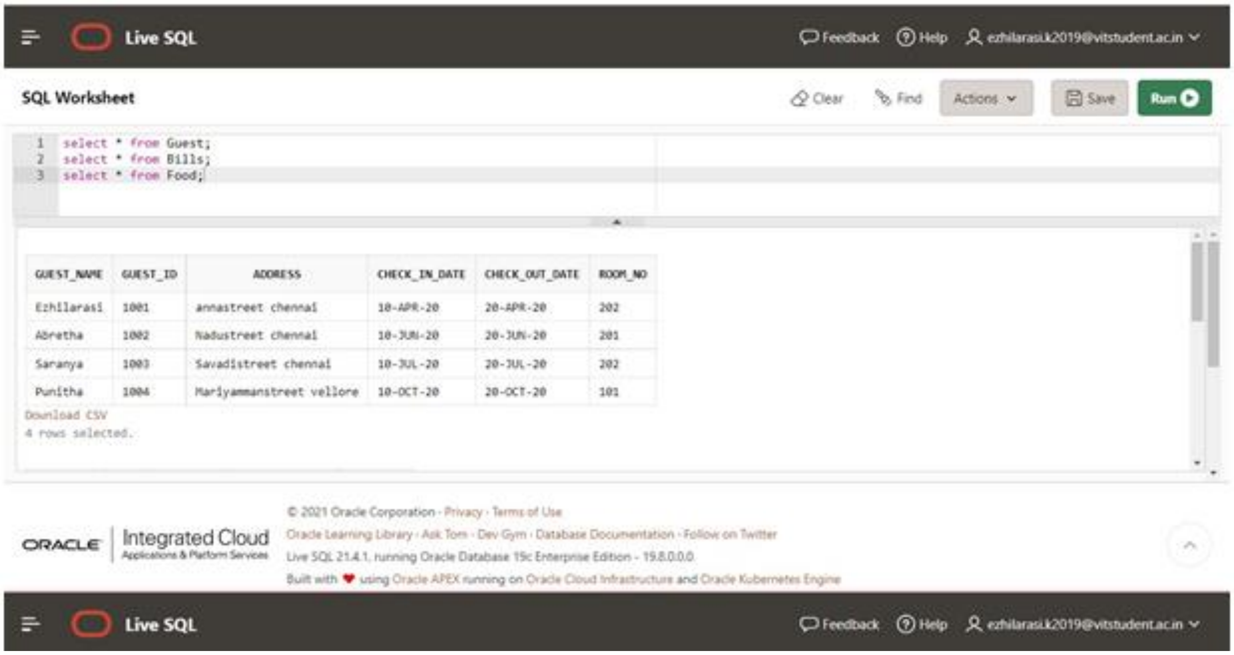


```sql
select * from Workers;
```

NAME	WORKERS_ID	ADDRESS	RESORT_ID
Karthi	1	vallarstreet vellore	VGP1001
kala	2	Madamstreet vellore	VGP1001
Ramesh	3	kannanstreet chennai	GCR2201
Ranjini	4	kamalastreet chennai	GCR2201

Download CSV
4 rows selected.

© 2021 Oracle Corporation · Privacy · Terms of Use
Oracle Learning Library · Ask Tom · Dev Gym · Database Documentation · Follow on Twitter
Live SQL 21.4.1, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0
Built with ❤ using Oracle APEX running on Oracle Cloud Infrastructure and Oracle Kubernetes Engine



```
select * from Guest;
select * from Bills;
select * from Food;
```

GUEST_NAME	GUEST_ID	ADDRESS	CHECK_IN_DATE	CHECK_OUT_DATE	ROOM_NO
Ezhilarasi	1001	annastreet chennai	10-APR-20	20-APR-20	202
Abretha	1002	Nadustreet chennai	10-JUN-20	20-JUN-20	201
Saranya	1003	Savadistreet chennai	10-JUL-20	20-JUL-20	202
Punitha	1004	Mariyammanstreet vellore	10-OCT-20	20-OCT-20	101

Download CSV

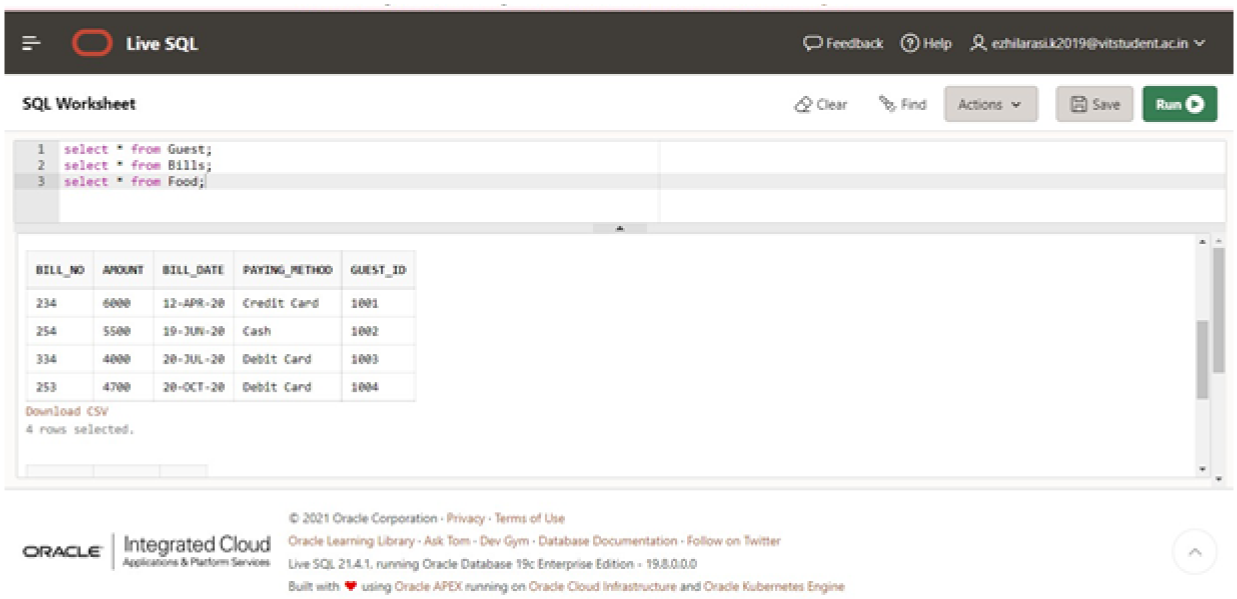
4 rows selected.

© 2021 Oracle Corporation · Privacy · Terms of Use

Oracle Learning Library · Ask Tom · Dev Gym · Database Documentation · Follow on Twitter

Live SQL 21.4.1, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0

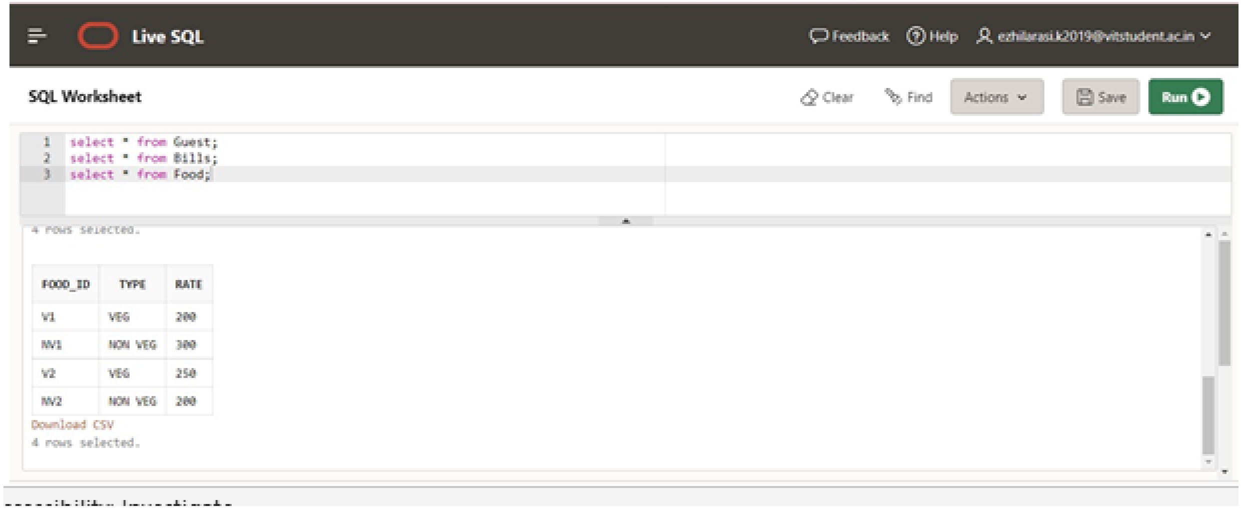
Built with ♥ using Oracle APEX running on Oracle Cloud Infrastructure and Oracle Kubernetes Engine


Live SQL
Feedback
Help
ezhilarasi.k2019@vitstudent.ac.in
SQL Worksheet
Clear
Find
Actions
Save
Run
Download CSV
ORACLE Integrated Cloud
Applications & Platform Services
© 2021 Oracle Corporation · Privacy · Terms of Use
Oracle Learning Library · Ask Tom · Dev Gym · Database Documentation · Follow on Twitter
Live SQL 21.4.1, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0
Built with ♥ using Oracle APEX running on Oracle Cloud Infrastructure and Oracle Kubernetes Engine

```
select * from Guest;
select * from Bills;
select * from Food;
```

BILL_NO	AMOUNT	BILL_DATE	PAYING_METHOD	GUEST_ID
234	6000	12-APR-20	Credit Card	1001
254	5500	19-JUN-20	Cash	1002
334	4000	20-JUL-20	Debit Card	1003
253	4700	20-OCT-20	Debit Card	1004

```
4 rows selected.
```

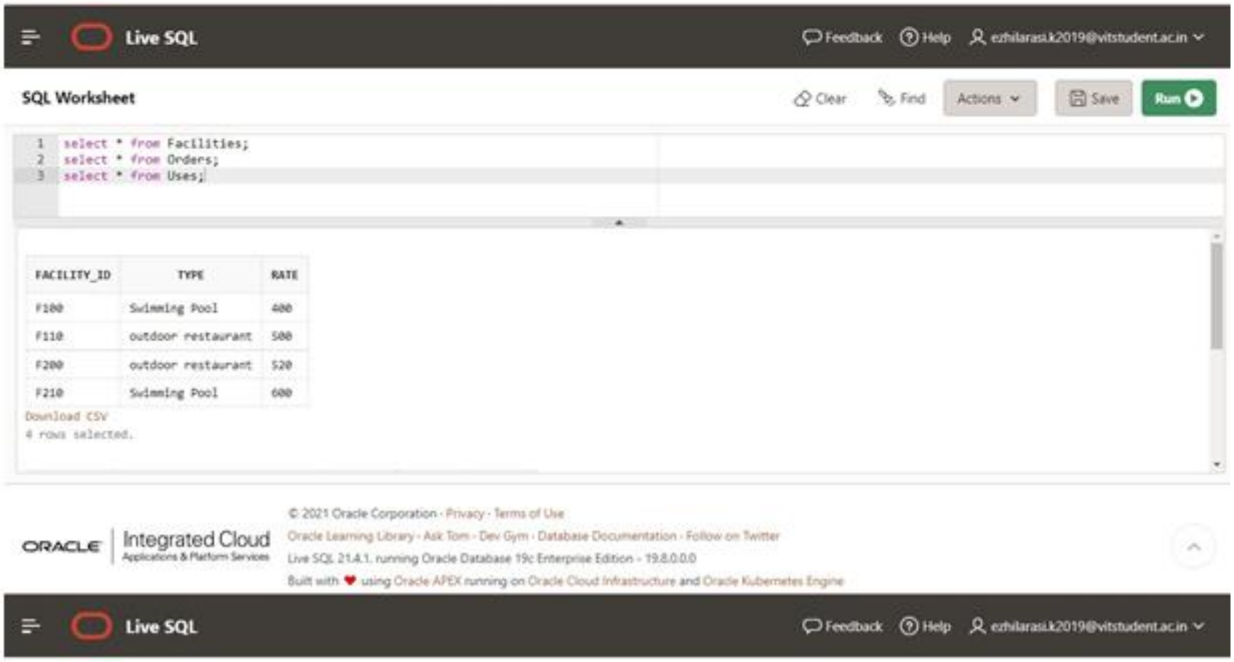


```
select * from Guest;
select * from Bills;
select * from Food;
```

```
4 rows selected.
```

FOOD_ID	TYPE	RATE
V1	VEG	200
NV1	NON VEG	300
V2	VEG	250
NV2	NON VEG	200

```
4 rows selected.
```



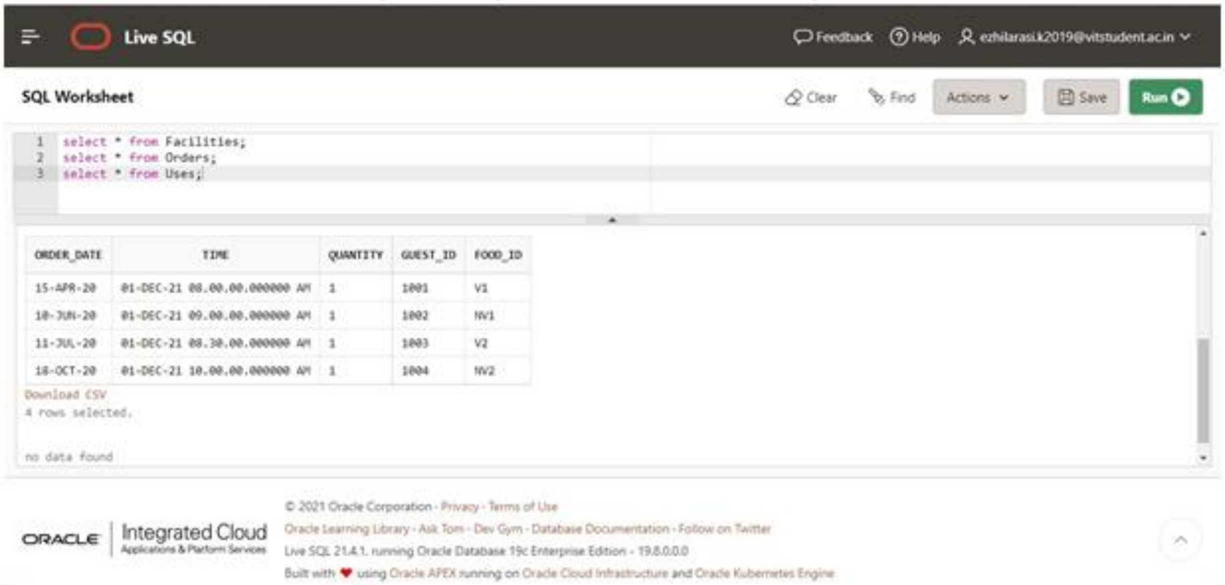
```
select * from Facilities;
select * from Orders;
select * from Uses;
```

FACILITY_ID	TYPE	RATE
F100	Swimming Pool	400
F110	outdoor restaurant	500
F200	outdoor restaurant	520
F210	Swimming Pool	600

Download CSV
4 rows selected.

© 2021 Oracle Corporation · Privacy · Terms of Use
Oracle Learning Library · Ask Tom · Dev Gym · Database Documentation · Follow on Twitter
Live SQL 21.4.1, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0
Built with ♥ using Oracle APEX running on Oracle Cloud Infrastructure and Oracle Kubernetes Engine

ORACLE Integrated Cloud Applications & Platform Services
Live SQL
Feedback
Help
ezhilarasi.k2019@vitstudent.ac.in


Live SQL
Feedback
Help
ezhilarasi.k2019@vitstudent.ac.in
SQL Worksheet
Clear
Find
Actions
Save
Run

```
select * from Facilities;
select * from Orders;
select * from Uses;
```

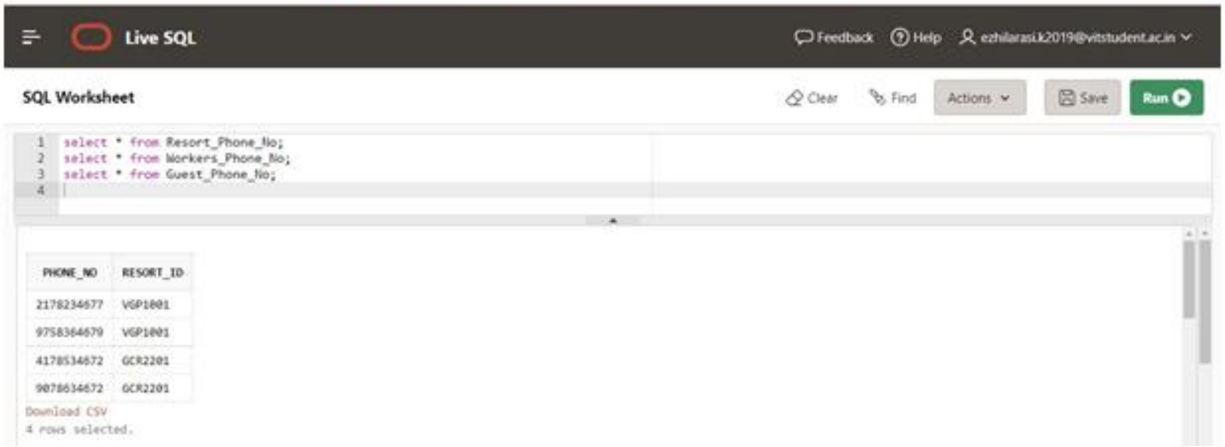
ORDER_DATE	TIME	QUANTITY	GUEST_ID	FOOD_ID
15-APR-20	01-DEC-21 08.00.00.000000 AM	1	1001	V1
10-JUN-20	01-DEC-21 09.00.00.000000 AM	1	1002	NV1
11-JUL-20	01-DEC-21 08.30.00.000000 AM	1	1003	V2
18-OCT-20	01-DEC-21 10.00.00.000000 AM	1	1004	NV2

Download CSV
4 rows selected.

no data found

© 2021 Oracle Corporation · Privacy · Terms of Use
Oracle Learning Library · Ask Tom · Dev Gym · Database Documentation · Follow on Twitter
Live SQL 21.4.1, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0
Built with ♥ using Oracle APEX running on Oracle Cloud Infrastructure and Oracle Kubernetes Engine

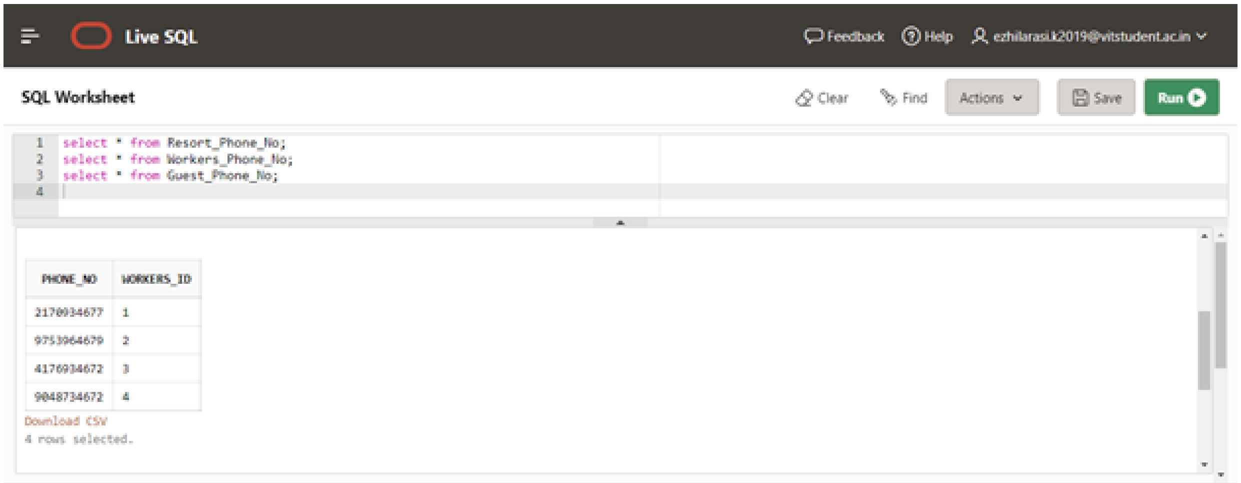
ORACLE Integrated Cloud Applications & Platform Services


Live SQL
Feedback
Help
ezhilarasi.k2019@vitstudent.ac.in
SQL Worksheet
Clear
Find
Actions
Save
Run

```
select * from Resort_Phone_No;
select * from Workers_Phone_No;
select * from Guest_Phone_No;
```

PHONE_NO	RESORT_ID
2178234677	VGP1001
9758364679	VGP1001
4178534672	GCR2201
9078634672	GCR2201

Download CSV
4 rows selected.

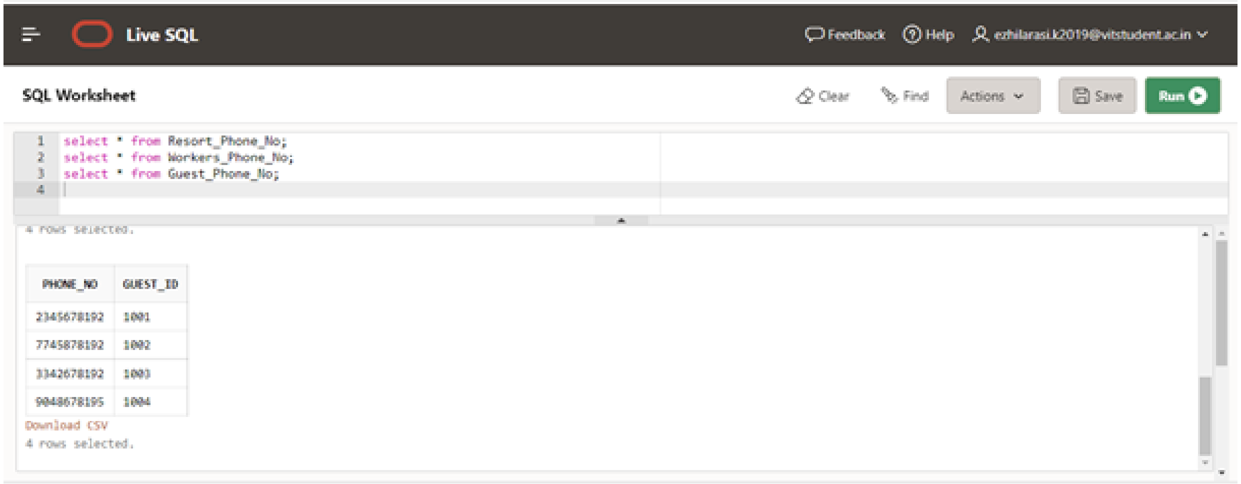


```
select * from Resort_Phone_No;
select * from Workers_Phone_No;
select * from Guest_Phone_No;
```

PHONE_NO	WORKERS_ID
2170934677	1
9753964679	2
4176934672	3
9048734672	4

Download CSV

4 rows selected.



```
select * from Resort_Phone_No;
select * from Workers_Phone_No;
select * from Guest_Phone_No;
```

4 rows selected.

PHONE_NO	GUEST_ID
2345678192	1001
7745878192	1002
3342678192	1003
9048678195	1004

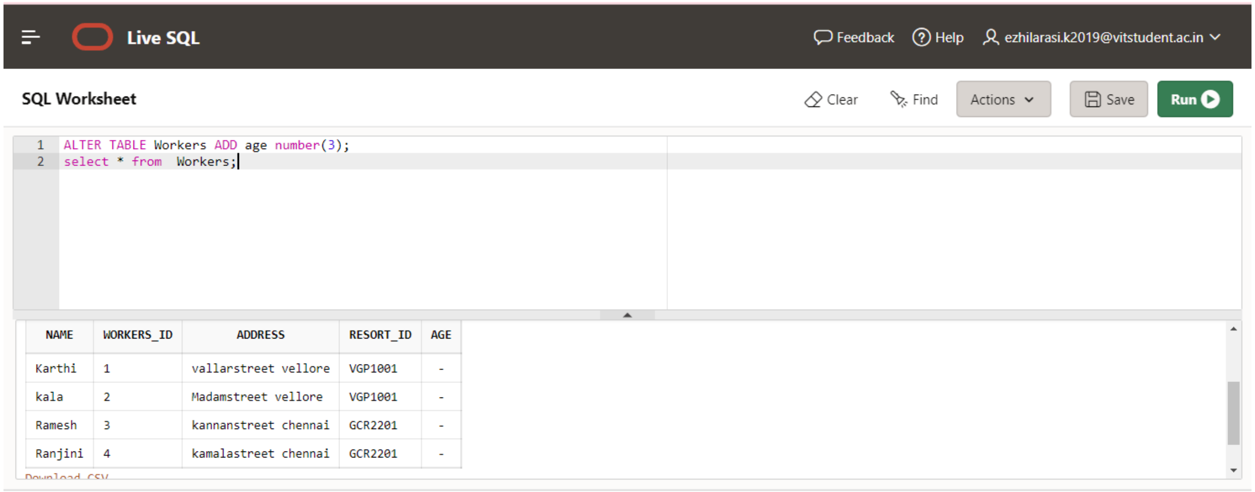
Download CSV

4 rows selected.

Alter ,delete and update:

ALTER TABLE - ADD Column

```
ALTER TABLE Workers ADD age number(3);
select * from  Workers;
```



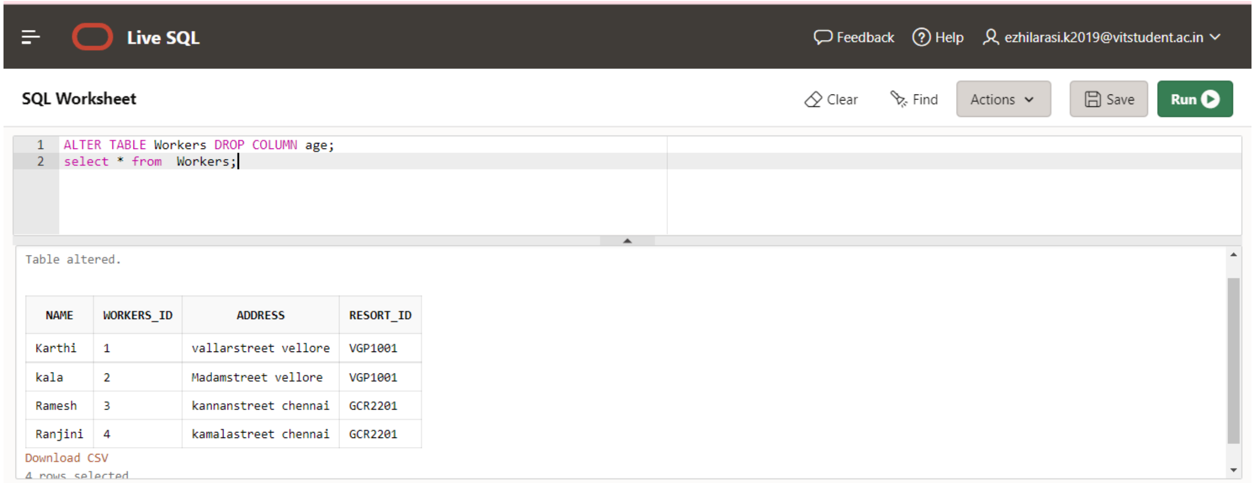
```
ALTER TABLE Workers ADD age number(3);
select * from  Workers;
```

NAME	WORKERS_ID	ADDRESS	RESORT_ID	AGE
Karthi	1	vallarstreet vellore	VGP1001	-
kala	2	Madamstreet vellore	VGP1001	-
Ramesh	3	kannanstreet chennai	GCR2201	-
Ranjini	4	kamalastreet chennai	GCR2201	-

Download CSV

ALTER TABLE - DROP COLUMN

ALTER TABLE Workers DROP COLUMN age;
select * from Workers;



```
ALTER TABLE Workers DROP COLUMN age;
select * from  Workers;
```

```
Table altered.
```

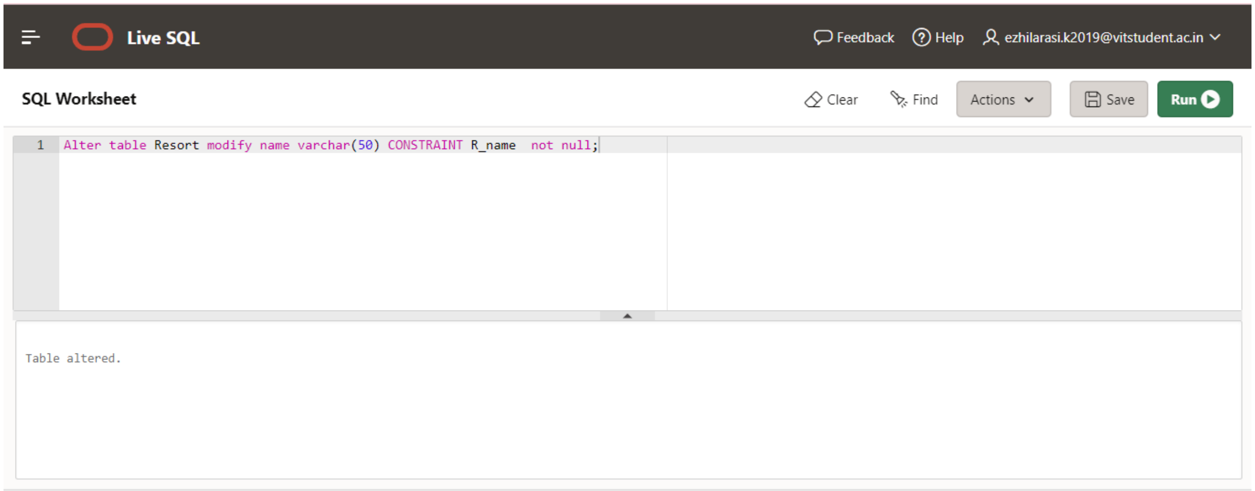
NAME	WORKERS_ID	ADDRESS	RESORT_ID
Karthi	1	vallarstreet vellore	VGP1001
kala	2	Madamstreet vellore	VGP1001
Ramesh	3	kannanstreet chennai	GCR2201
Ranjini	4	kamalastreet chennai	GCR2201

Download CSV
4 rows selected.

ALTER TABLE - ALTER/MODIFY COLUMN

NOT NULL CONSTRAINT:

Alter table Resort modify name varchar(50) CONSTRAINT R_name not null;

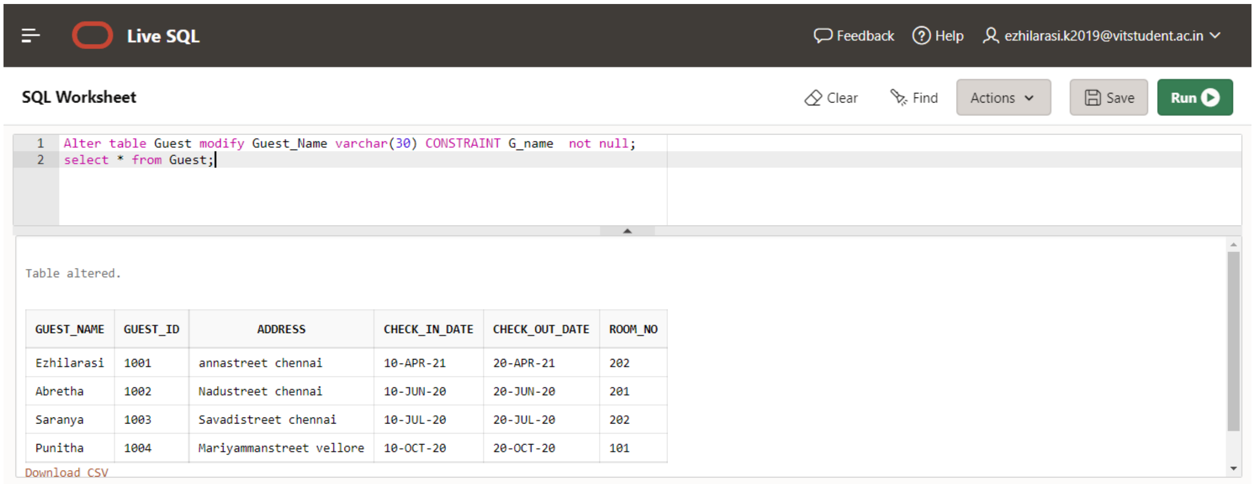


```
Alter table Resort modify name varchar(50) CONSTRAINT R_name  not null;
```

```
Table altered.
```

Alter table Guest modify Guest_Name varchar(30) CONSTRAINT G_name not null;

select * from Guest;



```
Alter table Guest modify Guest_Name varchar(30) CONSTRAINT G_name  not null;
select * from Guest;
```

```
Table altered.
```

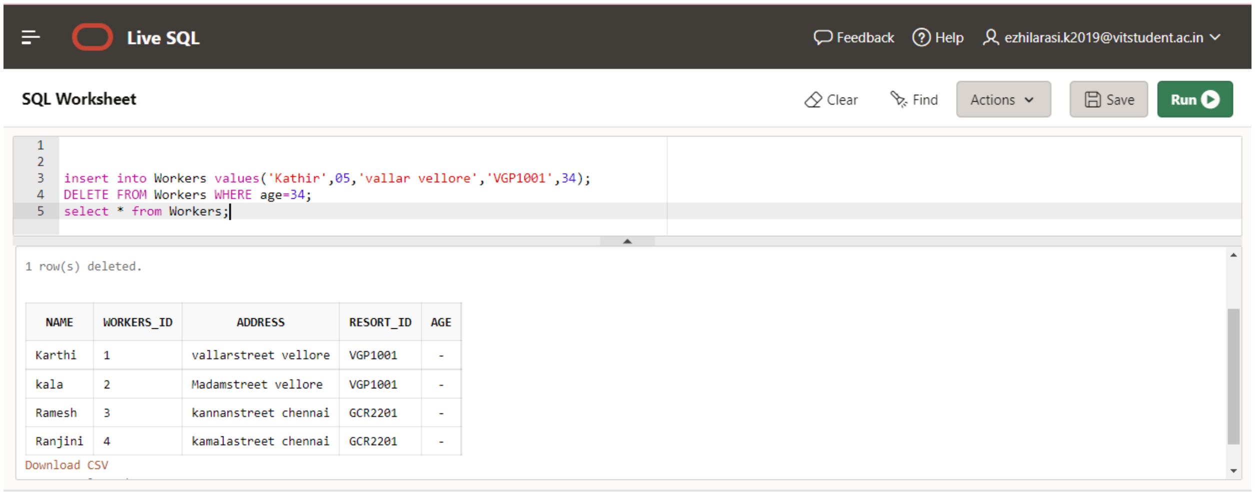
GUEST_NAME	GUEST_ID	ADDRESS	CHECK_IN_DATE	CHECK_OUT_DATE	ROOM_NO
Ezhilarasi	1001	annastreet chennai	10-APR-21	20-APR-21	202
Abretha	1002	Nadustreet chennai	10-JUN-20	20-JUN-20	201
Saranya	1003	Savadistreet chennai	10-JUL-20	20-JUL-20	202
Punitha	1004	Mariyammanstreet vellore	10-OCT-20	20-OCT-20	101

DELETE Statement

insert into Workers values('Kathir',05,'vallar vellore','VGP1001',34);

DELETE FROM Workers WHERE age=34;

select * from Workers;



```
insert into Workers values('Kathir',05,'vallar vellore','VGP1001',34);
DELETE FROM Workers WHERE age=34;
select * from Workers;
```

```
1 row(s) deleted.
```

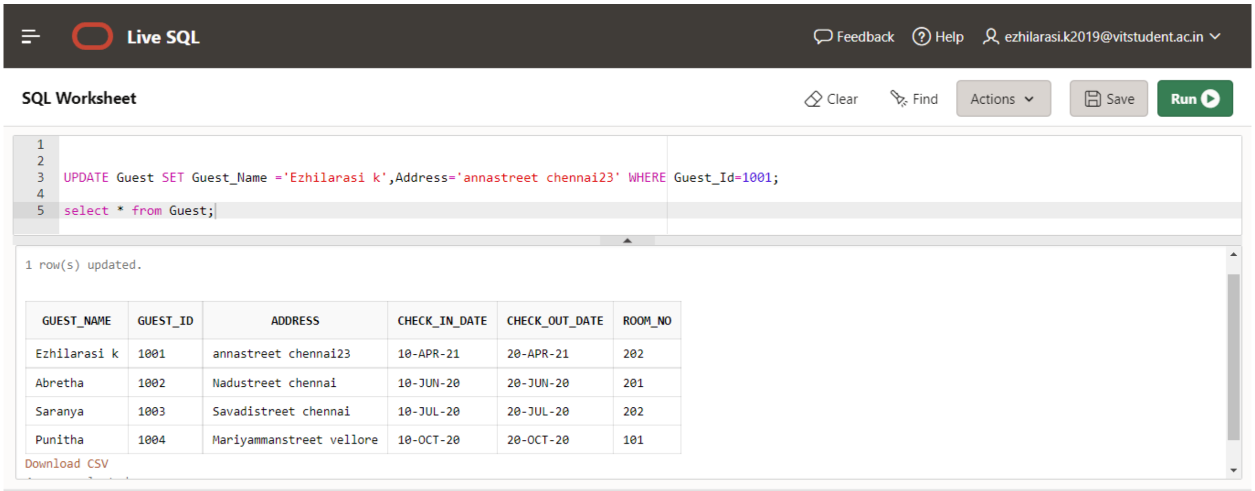
NAME	WORKERS_ID	ADDRESS	RESORT_ID	AGE
Karthi	1	vallarstreet vellore	VGP1001	-
kala	2	Madamstreet vellore	VGP1001	-
Ramesh	3	kannanstreet chennai	GCR2201	-
Ranjini	4	kamalastreet chennai	GCR2201	-

Download CSV

UPDATE Statement

UPDATE Guest SET Guest_Name ='Ezhilarasi k',Address='annastreet chennai23' WHERE Guest_Id=1001;

select * from Guest;



```
UPDATE Guest SET Guest_Name ='Ezhilarasi k',Address='annastreet chennai23' WHERE Guest_Id=1001;

select * from Guest;
```

```
1 row(s) updated.
```

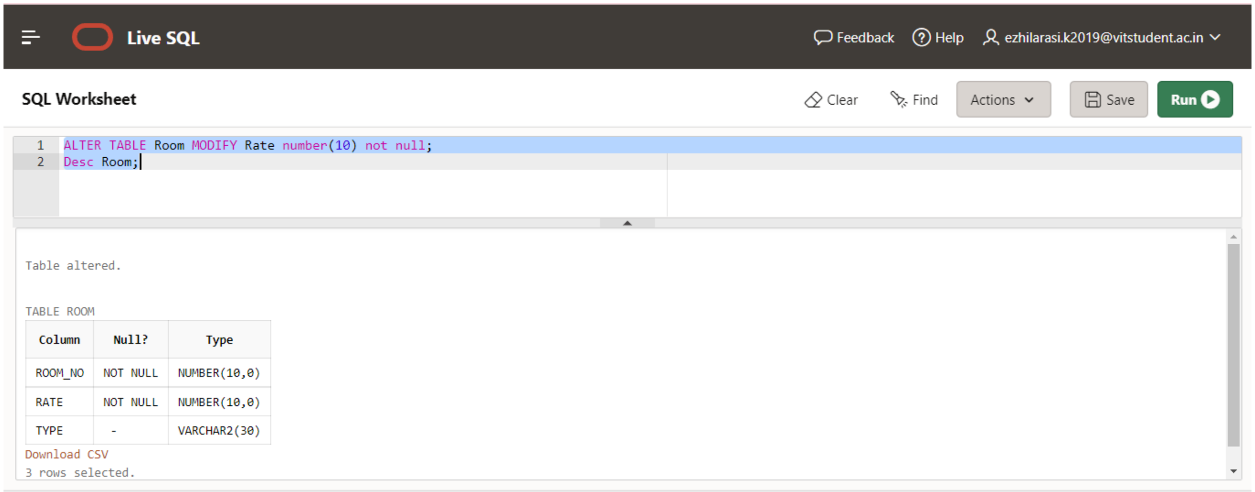
GUEST_NAME	GUEST_ID	ADDRESS	CHECK_IN_DATE	CHECK_OUT_DATE	ROOM_NO
Ezhilarasi k	1001	annastreet chennai23	10-APR-21	20-APR-21	202
Abretha	1002	Nadustreet chennai	10-JUN-20	20-JUN-20	201
Saranya	1003	Savadistreet chennai	10-JUL-20	20-JUL-20	202
Punitha	1004	Mariyammanstreet vellore	10-OCT-20	20-OCT-20	101

Download CSV

Constraint:

NOT NULL on ALTER TABLE

ALTER TABLE Room MODIFY Rate number(10) not null;
Desc Room;



```
ALTER TABLE Room MODIFY Rate number(10) not null;
Desc Room;
```

```
Table altered.

TABLE ROOM
```

Column	Null?	Type
ROOM_NO	NOT NULL	NUMBER(10,0)
RATE	NOT NULL	NUMBER(10,0)
TYPE	-	VARCHAR2(30)

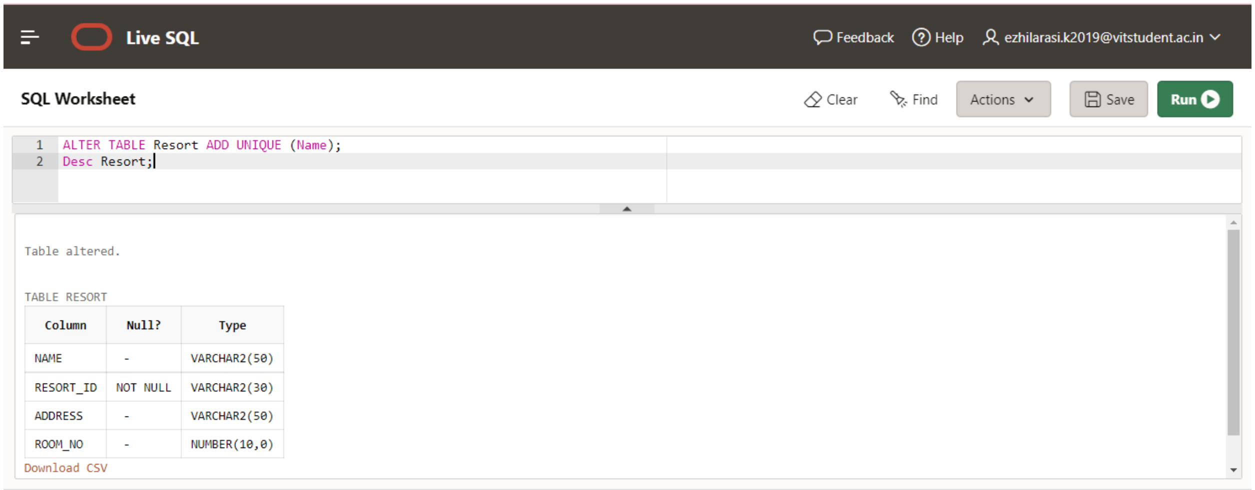
Download CSV

3 rows selected.

Unique:

ALTER TABLE Resort ADD UNIQUE (Name);

Desc Resort;



```
ALTER TABLE Resort ADD UNIQUE (Name);
Desc Resort;
```

```
Table altered.

TABLE RESORT
```

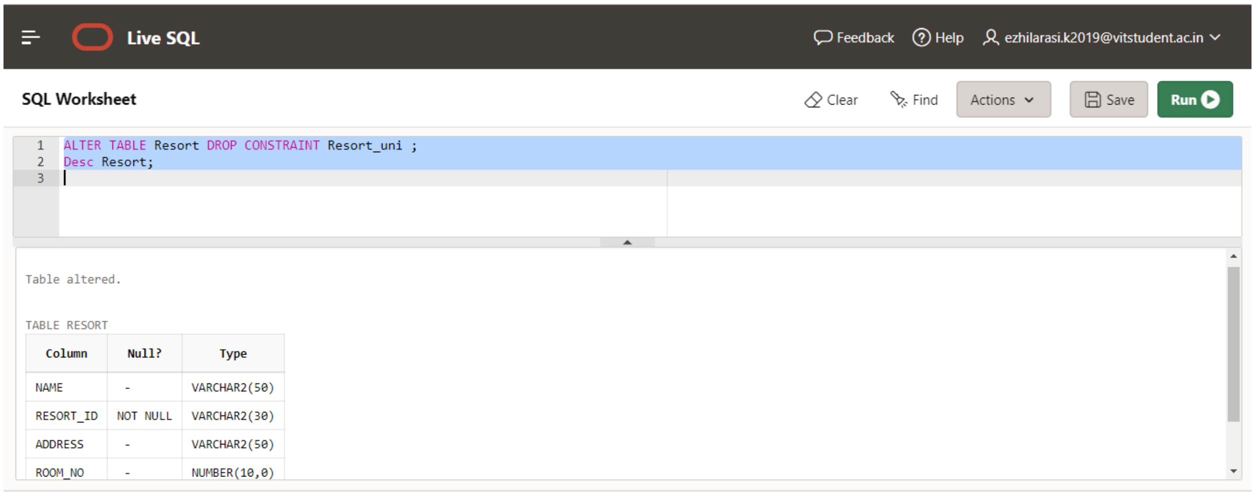
Column	Null?	Type
NAME	-	VARCHAR2(50)
RESORT_ID	NOT NULL	VARCHAR2(30)
ADDRESS	-	VARCHAR2(50)
ROOM_NO	-	NUMBER(10,0)

Download CSV

Drop unique:

ALTER TABLE Resort DROP CONSTRAINT Resort_uni ;

Desc Resort;



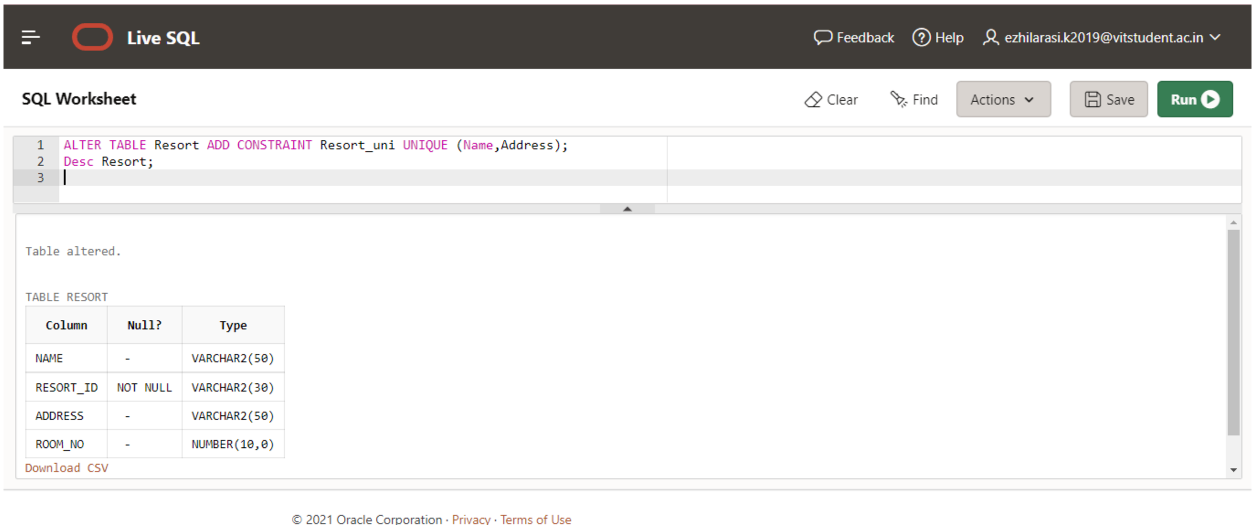
```
ALTER TABLE Resort DROP CONSTRAINT Resort_uni ;
Desc Resort;
```

```
Table altered.

TABLE RESORT
```

Column	Null?	Type
NAME	-	VARCHAR2(50)
RESORT_ID	NOT NULL	VARCHAR2(30)
ADDRESS	-	VARCHAR2(50)
ROOM_NO	-	NUMBER(10,0)

ALTER TABLE Resort ADD CONSTRAINT Resort_uni UNIQUE (Name,Address);

Desc Resort;



```
ALTER TABLE Resort ADD CONSTRAINT Resort_uni UNIQUE (Name,Address);
Desc Resort;
```

```
Table altered.

TABLE RESORT
```

Column	Null?	Type
NAME	-	VARCHAR2(50)
RESORT_ID	NOT NULL	VARCHAR2(30)
ADDRESS	-	VARCHAR2(50)
ROOM_NO	-	NUMBER(10,0)

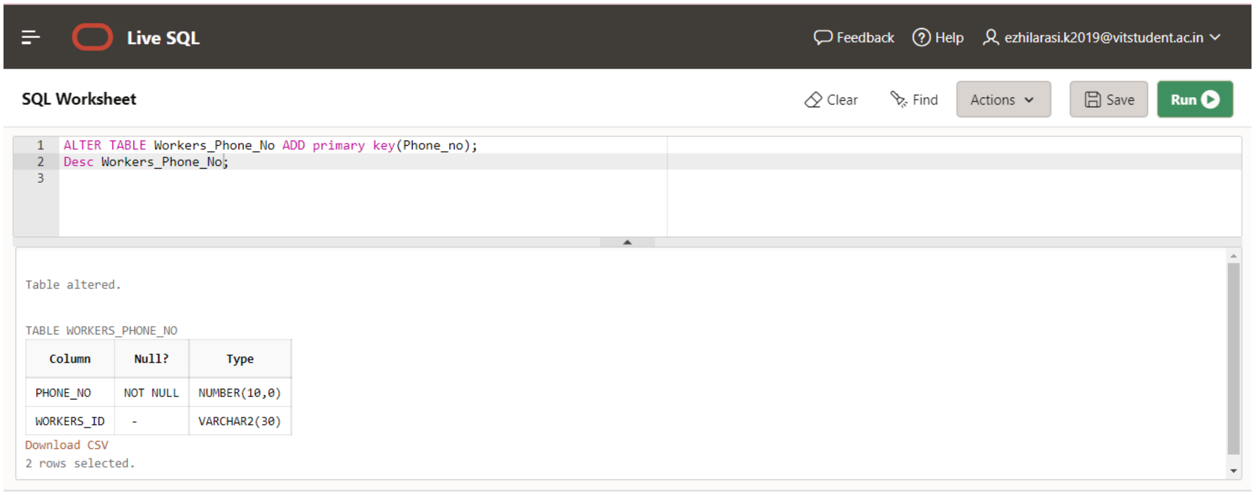
Download CSV

© 2021 Oracle Corporation · Privacy · Terms of Use

PRIMARY KEY:

ALTER TABLE Workers_Phone_No ADD primary key(Phone_no);

Desc Workers_Phone_No;



```
ALTER TABLE Workers_Phone_No ADD primary key(Phone_no);
Desc Workers_Phone_No;
```

```
Table altered.

TABLE WORKERS_PHONE_NO
```

Column	Null?	Type
PHONE_NO	NOT NULL	NUMBER(10,0)
WORKERS_ID	-	VARCHAR2(30)

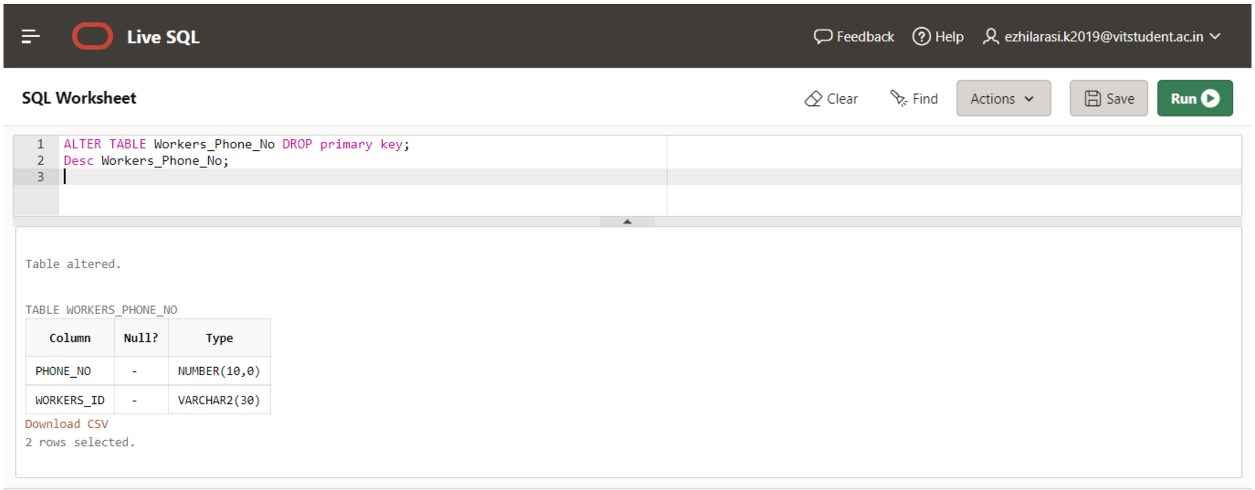
Download CSV

2 rows selected.

Drop pk:

ALTER TABLE Workers_Phone_No DROP primary key;

Desc Workers_Phone_No;



```
ALTER TABLE Workers_Phone_No DROP primary key;
Desc Workers_Phone_No;
```

```
Table altered.

TABLE WORKERS_PHONE_NO
```

Column	Null?	Type
PHONE_NO	-	NUMBER(10,0)
WORKERS_ID	-	VARCHAR2(30)

Download CSV

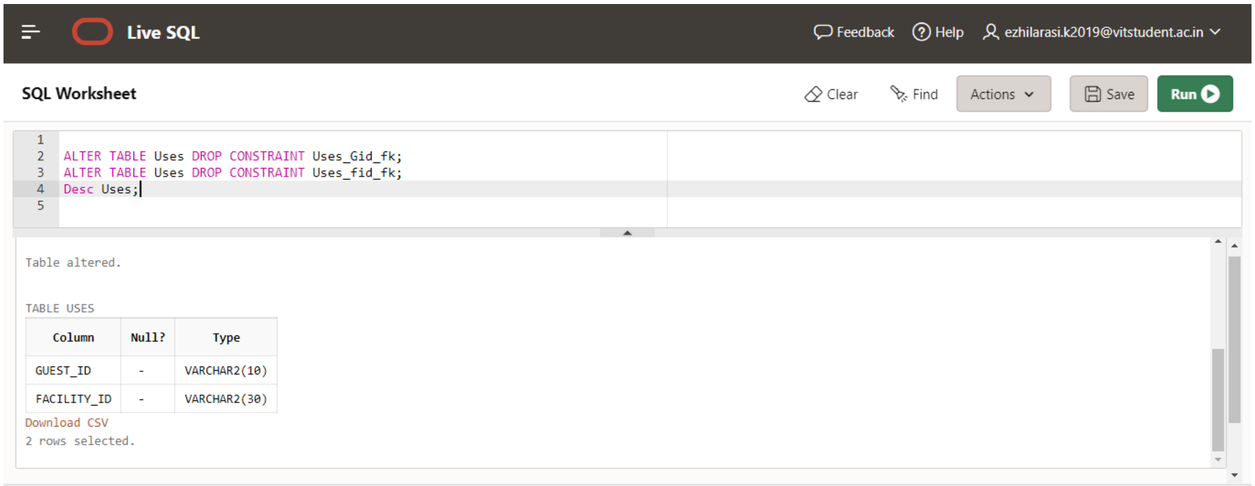
2 rows selected.

Fk drop:

ALTER TABLE Uses DROP CONSTRAINT Uses_Gid_fk;

ALTER TABLE Uses DROP CONSTRAINT Uses_fid_fk;

Desc Uses;



```
ALTER TABLE Uses DROP CONSTRAINT Uses_Gid_fk;
ALTER TABLE Uses DROP CONSTRAINT Uses_fid_fk;
Desc Uses;
```

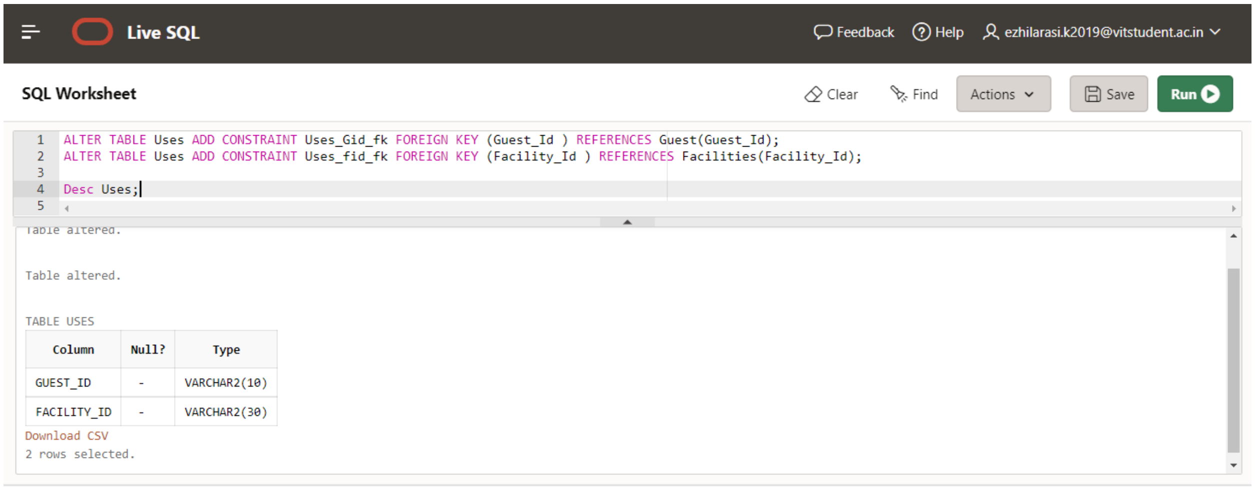
```
Table altered.

TABLE USES
Column        Null?   Type
GUEST_ID      -       VARCHAR2(10)
FACILITY_ID   -       VARCHAR2(30)
Download CSV
2 rows selected.
```

Add fk:

ALTER TABLE Uses ADD CONSTRAINT Uses_Gid_fk FOREIGN KEY (Guest_Id) REFERENCES Guest(Guest_Id);

ALTER TABLE Uses ADD CONSTRAINT Uses_fid_fk FOREIGN KEY (Facility_Id) REFERENCES Facilities(Facility_Id);

Desc Uses;



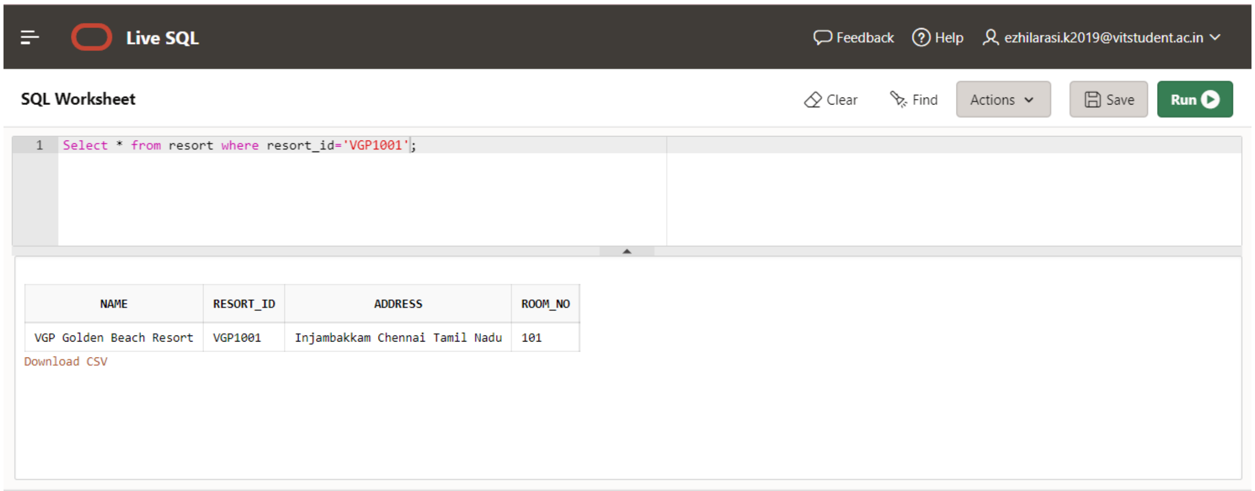
```
ALTER TABLE Uses ADD CONSTRAINT Uses_Gid_fk FOREIGN KEY (Guest_Id ) REFERENCES Guest(Guest_Id);
ALTER TABLE Uses ADD CONSTRAINT Uses_fid_fk FOREIGN KEY (Facility_Id ) REFERENCES Facilities(Facility_Id);

Desc Uses;
```

```
Table altered.

Table altered.

TABLE USES
Column        Null?   Type
GUEST_ID      -       VARCHAR2(10)
FACILITY_ID   -       VARCHAR2(30)
Download CSV
2 rows selected.
```

Select with where clause:

1.display the resort details which have a id ‘VGP1001’.

Select * from resort where resort_id=’VGP1001’;



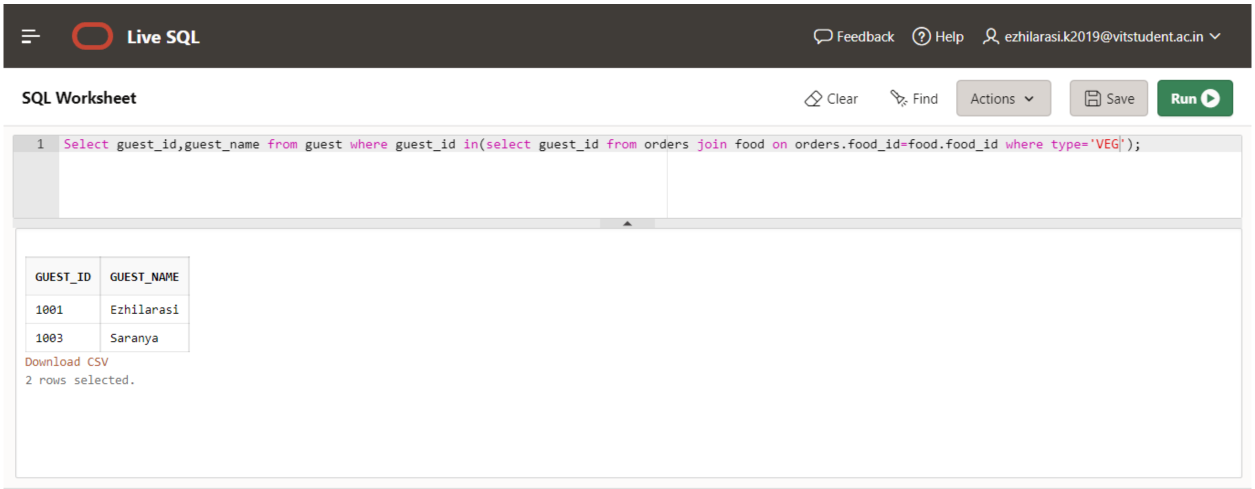
```
Select * from resort where resort_id='VGP1001';
```

NAME	RESORT_ID	ADDRESS	ROOM_NO
VGP Golden Beach Resort	VGP1001	Injambakkam Chennai Tamil Nadu	101

Download CSV

Nested query:

display the guest_name,guest_id who is orders veg food using nested query.
Select guest_id,guest_name from guest where guest_id in(select guest_id from orders join food on orders.food_id=food.food_id where type='VEG');



```
Select guest_id,guest_name from guest where guest_id in(select guest_id from orders join food on orders.food_id=food.food_id where type='VEG');
```

GUEST_ID	GUEST_NAME
1001	Ezhilarasi
1003	Saranya

Download CSV
2 rows selected.

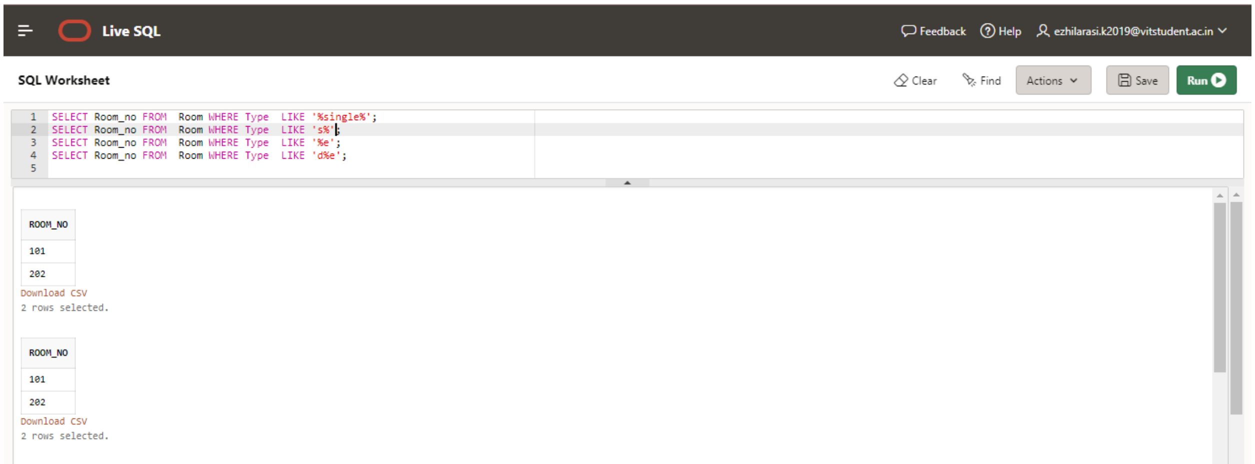
Like clause:

SELECT Room_no FROM Room WHERE Type LIKE '%single%';

SELECT Room_no FROM Room WHERE Type LIKE 's%';

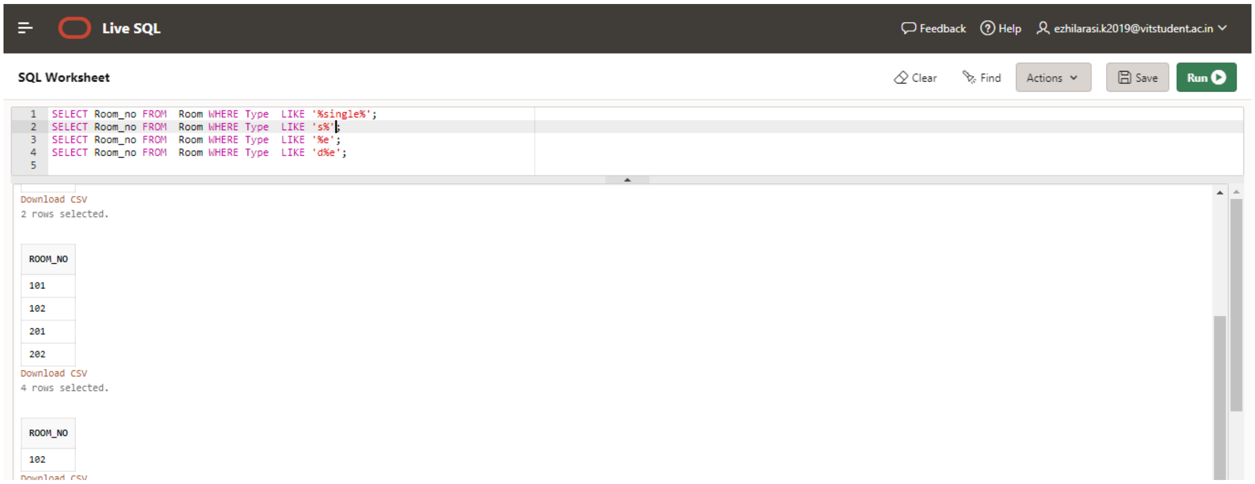
SELECT Room_no FROM Room WHERE Type LIKE '%e';

SELECT Room_no FROM Room WHERE Type LIKE 'd%e';



```
SELECT Room_no FROM  Room WHERE Type  LIKE '%single%';
SELECT Room_no FROM  Room WHERE Type  LIKE 's%';
SELECT Room_no FROM  Room WHERE Type  LIKE '%e';
SELECT Room_no FROM  Room WHERE Type  LIKE 'd%e';
```

```
ROOM_NO
101
202
Download CSV
2 rows selected.

ROOM_NO
101
202
Download CSV
2 rows selected.
```



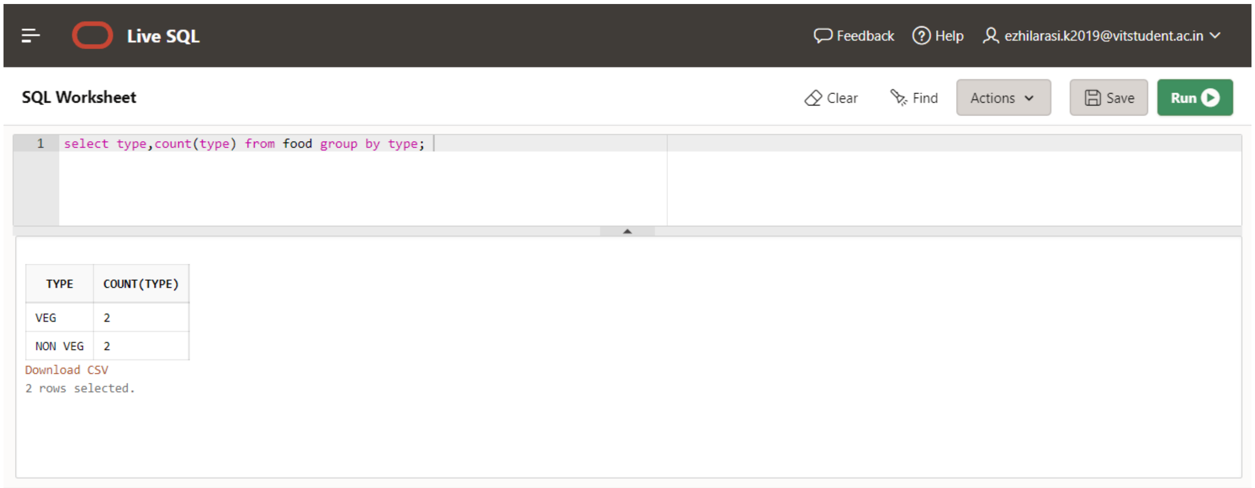
```
SELECT Room_no FROM  Room WHERE Type  LIKE '%single%';
SELECT Room_no FROM  Room WHERE Type  LIKE 's%';
SELECT Room_no FROM  Room WHERE Type  LIKE '%e';
SELECT Room_no FROM  Room WHERE Type  LIKE 'd%e';
```

```
Download CSV
2 rows selected.

ROOM_NO
101
102
201
202
Download CSV
4 rows selected.

ROOM_NO
102
Download CSV
```

Group by and having

display the food type with no of types for that particular type of food.

select type,count(type) from food group by type;

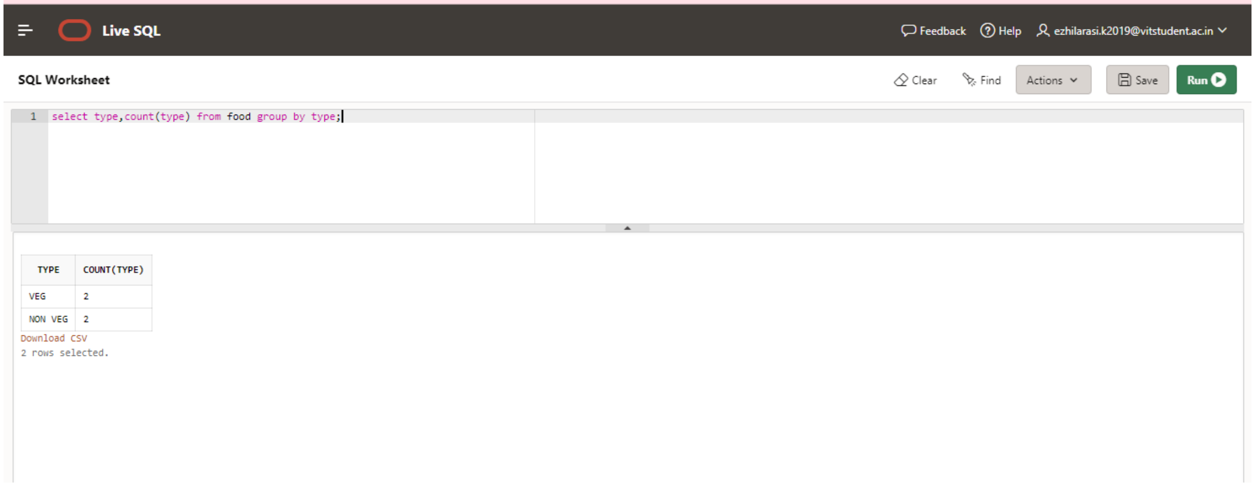


```
select type,count(type) from food group by type;
```

TYPE	COUNT(TYPE)
VEG	2
NON VEG	2

Download CSV
2 rows selected.

select type,count(type) from food group by type;

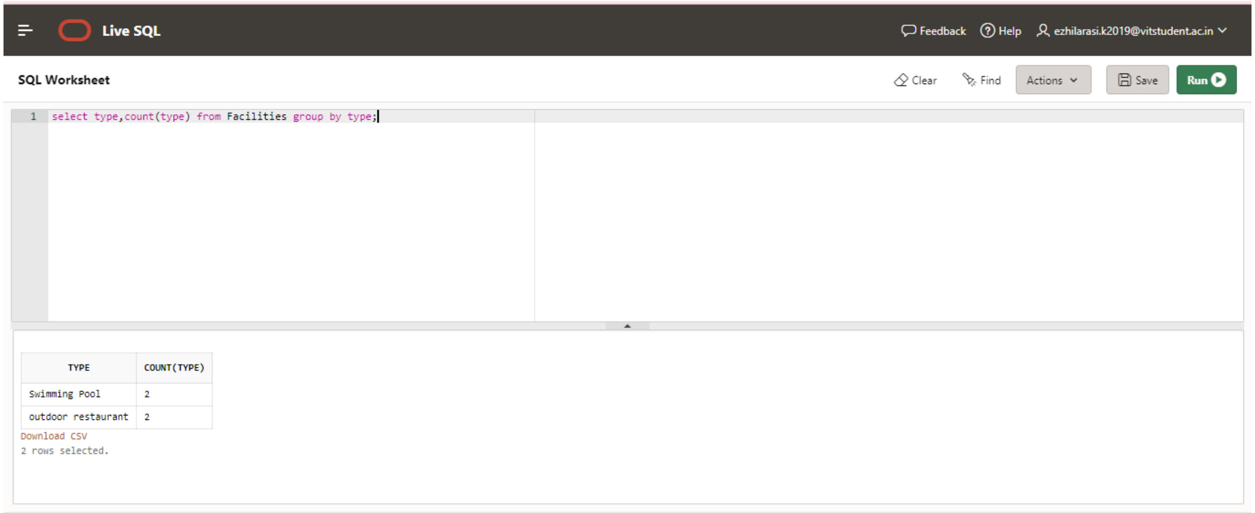


```
select type,count(type) from food group by type;
```

TYPE	COUNT(TYPE)
VEG	2
NON VEG	2

Download CSV
2 rows selected.

select type,count(type) from Facilities group by type;



```
select type,count(type) from Facilities group by type;
```

TYPE	COUNT(TYPE)
Swimming Pool	2
outdoor restaurant	2

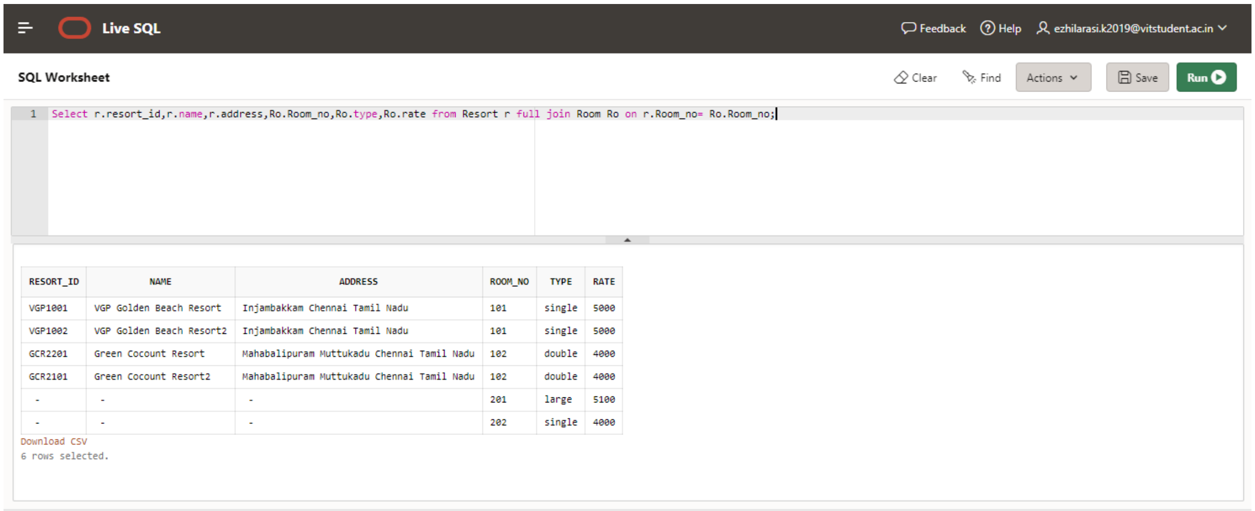
Download CSV
2 rows selected.

Join more than two tables

Full join:

display the room and resort details(Using left/right/full outer join query).

Select r.resort_id,r.name,r.address,Ro.Room_no,Ro.type,Ro.rate from Resort r full join Room Ro on r.Room_no= Ro.Room_no;



```
Select r.resort_id,r.name,r.address,Ro.Room_no,Ro.type,Ro.rate from Resort r full join Room Ro on r.Room_no= Ro.Room_no;
```

RESORT_ID	NAME	ADDRESS	ROOM_NO	TYPE	RATE
VGP1001	VGP Golden Beach Resort	Injambakkam Chennai Tamil Nadu	101	single	5000
VGP1002	VGP Golden Beach Resort2	Injambakkam Chennai Tamil Nadu	101	single	5000
GCR2201	Green Cocount Resort	Mahabalipuram Muttukadu Chennai Tamil Nadu	102	double	4000
GCR2101	Green Cocount Resort2	Mahabalipuram Muttukadu Chennai Tamil Nadu	102	double	4000
-	-	-	201	large	5100
-	-	-	202	single	4000

Download CSV
6 rows selected.

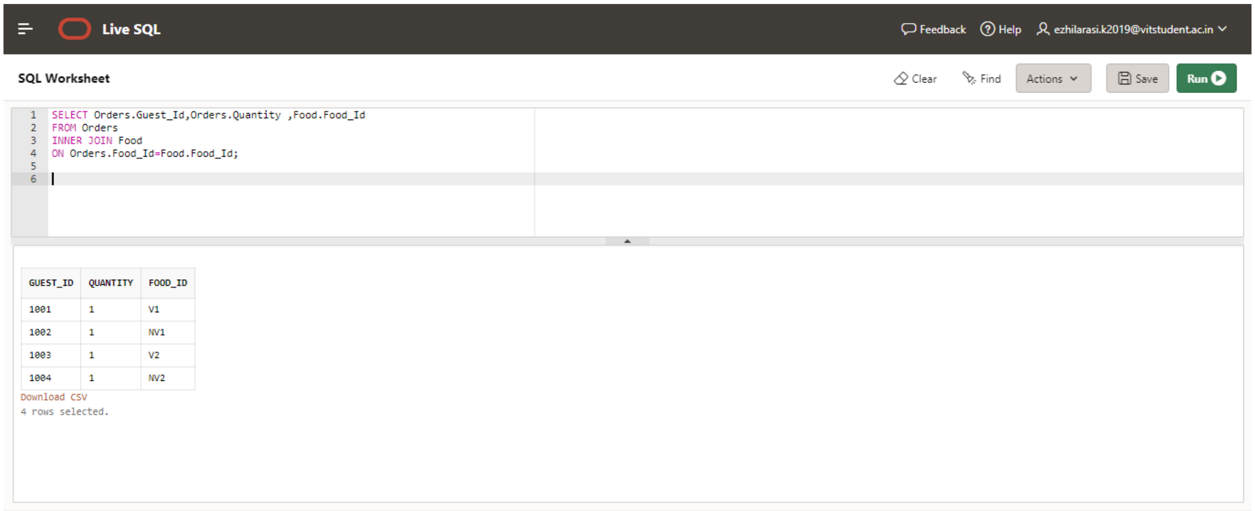
Inner join:

SELECT Orders.Guest_Id,Orders.Quantity ,Food.Food_Id

FROM Orders

INNER JOIN Food

ON Orders.Food_Id=Food.Food_Id;



```sql
SELECT Orders.Guest_Id,Orders.Quantity ,Food.Food_Id
FROM Orders
INNER JOIN Food
ON Orders.Food_Id=Food.Food_Id;
```

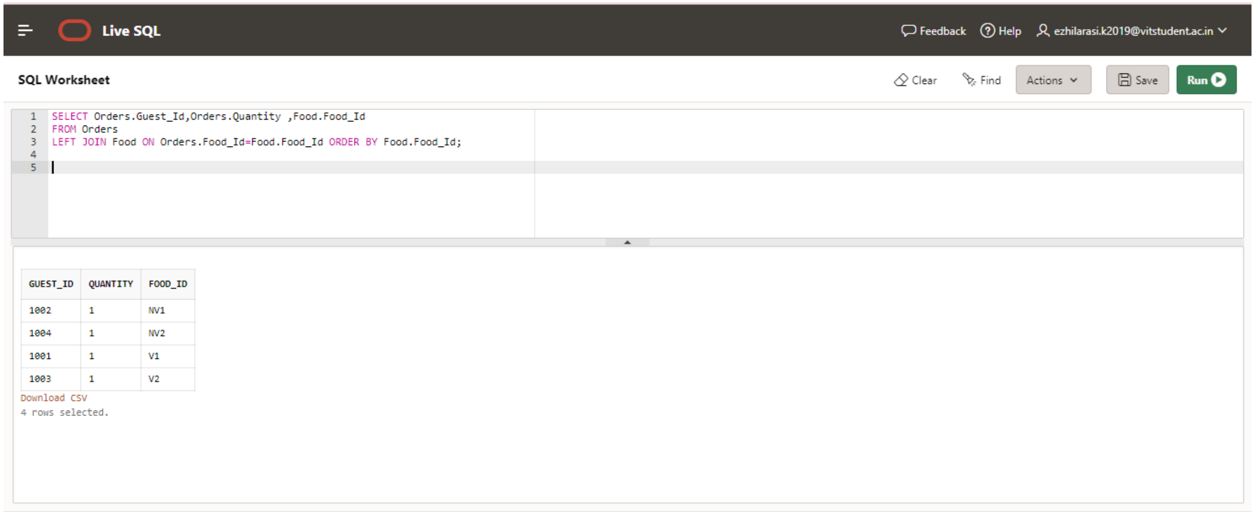
GUEST_ID	QUANTITY	FOOD_ID
1001	1	V1
1002	1	NV1
1003	1	V2
1004	1	NV2

Download CSV
4 rows selected.

SELECT Orders.Guest_Id,Orders.Quantity ,Food.Food_Id

FROM Orders

LEFT JOIN Food ON Orders.Food_Id=Food.Food_Id ORDER BY Food.Food_Id;



```sql
SELECT Orders.Guest_Id,Orders.Quantity ,Food.Food_Id
FROM Orders
LEFT JOIN Food ON Orders.Food_Id=Food.Food_Id ORDER BY Food.Food_Id;
```

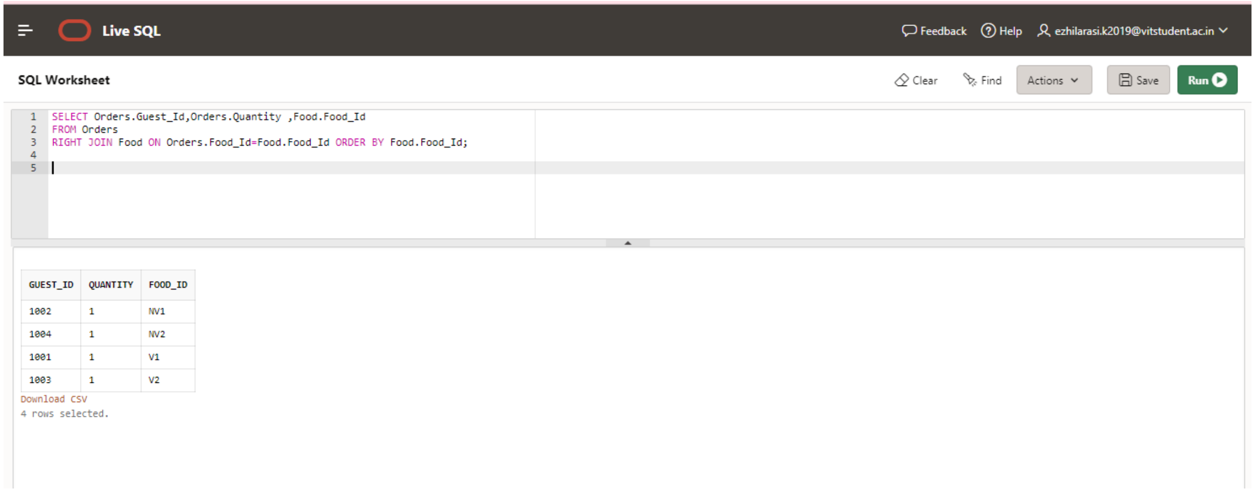
GUEST_ID	QUANTITY	FOOD_ID
1002	1	NV1
1004	1	NV2
1001	1	V1
1003	1	V2

Download CSV
4 rows selected.

SELECT Orders.Guest_Id,Orders.Quantity ,Food.Food_Id

FROM Orders

RIGHT JOIN Food ON Orders.Food_Id=Food.Food_Id ORDER BY Food.Food_Id;



```sql
SELECT Orders.Guest_Id,Orders.Quantity ,Food.Food_Id
FROM Orders
RIGHT JOIN Food ON Orders.Food_Id=Food.Food_Id ORDER BY Food.Food_Id;
```

GUEST_ID	QUANTITY	FOOD_ID
1002	1	NV1
1004	1	NV2
1001	1	V1
1003	1	V2

Download CSV

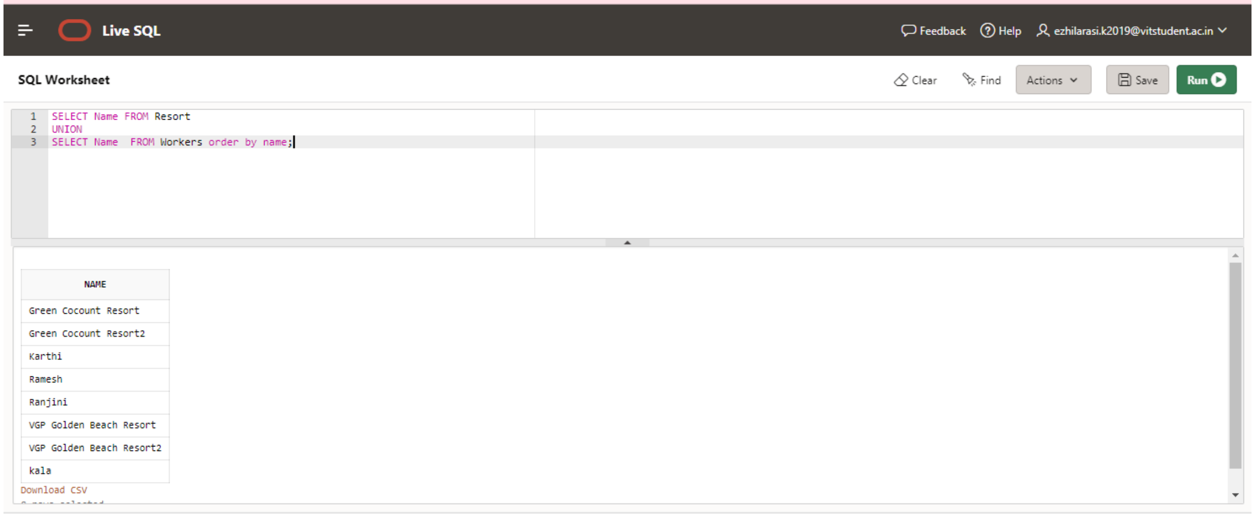
4 rows selected.

Union:

SELECT Name FROM Resort

UNION

SELECT Name FROM Workers order by name;



```sql
SELECT Name FROM Resort
UNION
SELECT Name  FROM Workers order by name;
```

NAME
Green Cocount Resort
Green Cocount Resort2
Karthi
Ramesh
Ranjini
VGP Golden Beach Resort
VGP Golden Beach Resort2
kala

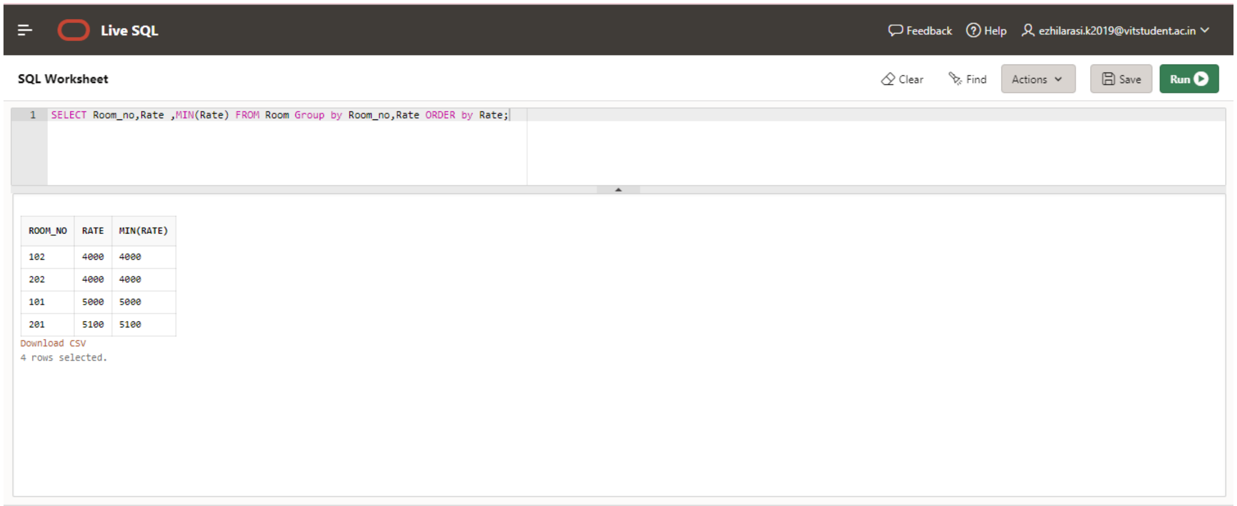
Download CSV

8 rows selected.

AGGREGATE FUNCTIONS:

Min()

SELECT Room_no,Rate ,MIN(Rate) FROM Room Group by Room_no,Rate ORDER by Rate



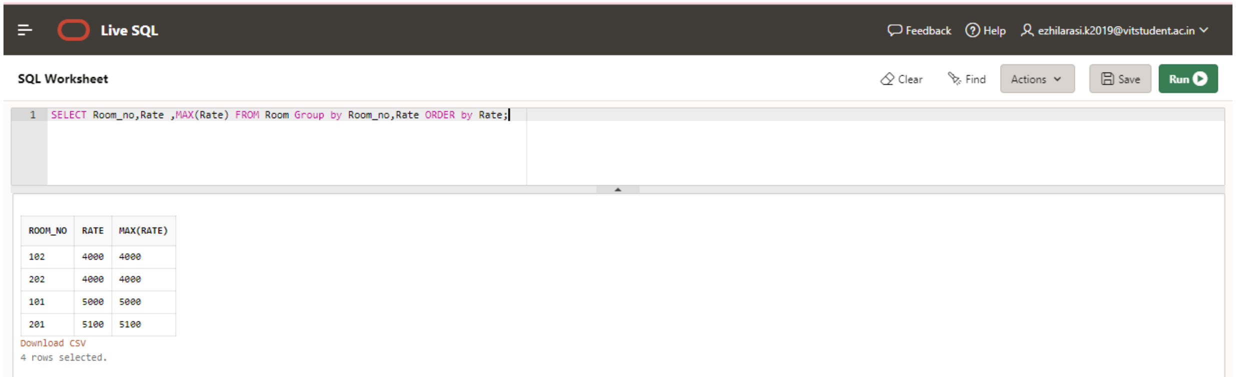
```
SELECT Room_no,Rate ,MIN(Rate) FROM Room Group by Room_no,Rate ORDER by Rate;
```

ROOM_NO	RATE	MIN(RATE)
102	4000	4000
202	4000	4000
101	5000	5000
201	5100	5100

```
Download CSV
4 rows selected.
```

MAX()

SELECT Room_no,Rate ,MAX(Rate) FROM Room Group by Room_no,Rate ORDER by Rate;



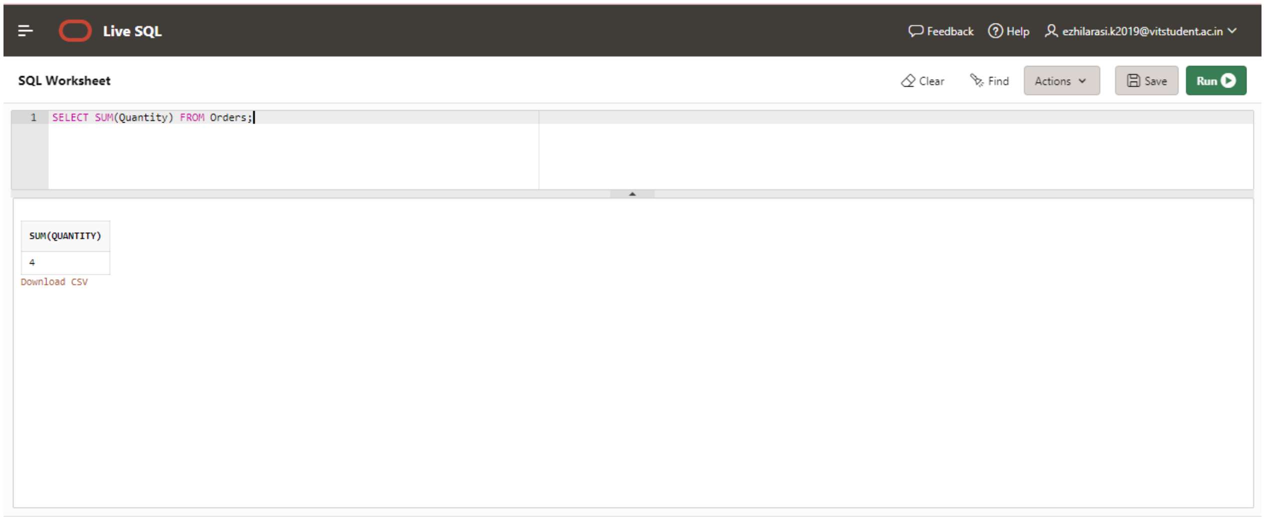
```
SELECT Room_no,Rate ,MAX(Rate) FROM Room Group by Room_no,Rate ORDER by Rate;
```

ROOM_NO	RATE	MAX(RATE)
102	4000	4000
202	4000	4000
101	5000	5000
201	5100	5100

```
Download CSV
4 rows selected.
```

Sum()

SELECT SUM(Quantity) FROM Orders;



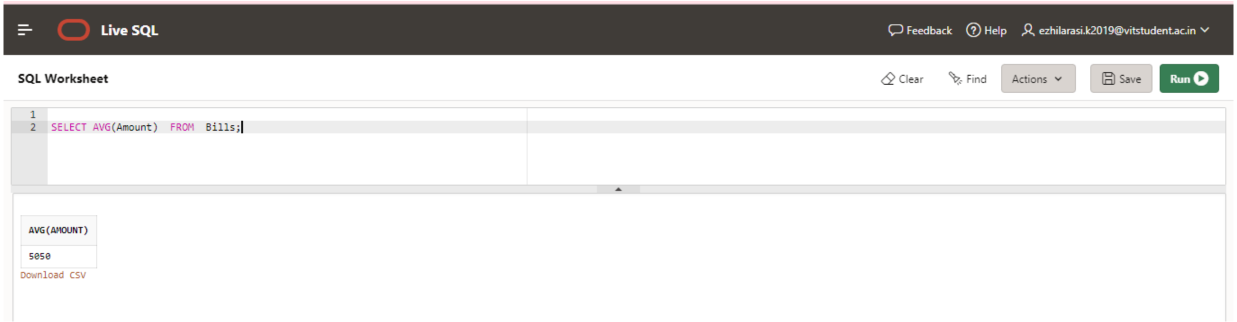
```
SELECT SUM(Quantity) FROM Orders;
```

SUM(QUANTITY)
4

```
Download CSV
```

Avg()

SELECT AVG(Amount) FROM Bills;



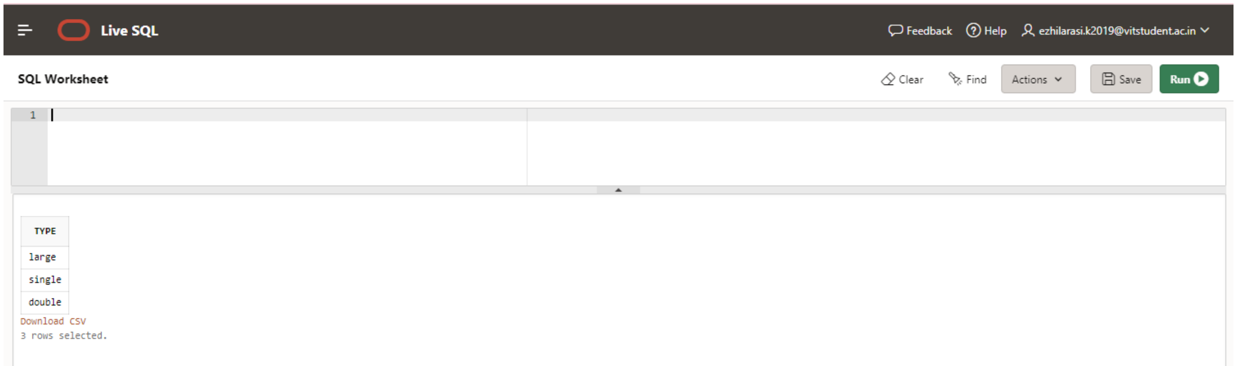
```

SELECT AVG(Amount)  FROM  Bills;
```

AVG(AMOUNT)
5050

Download CSV

Distinct:

SELECT distinct type FROM Room;



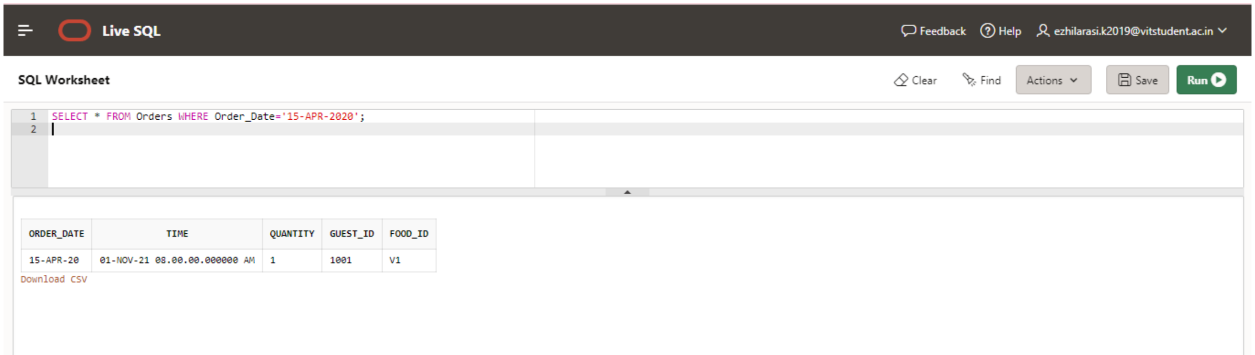
TYPE
large
single
double

Download CSV

3 rows selected.

Date functions:

SELECT * FROM Orders WHERE Order_Date='15-APR-2020';



```
SELECT * FROM Orders WHERE Order_Date='15-APR-2020';

```

ORDER_DATE	TIME	QUANTITY	GUEST_ID	FOOD_ID
15-APR-20	01-NOV-21 08.00.00.000000 AM	1	1001	V1

Download CSV

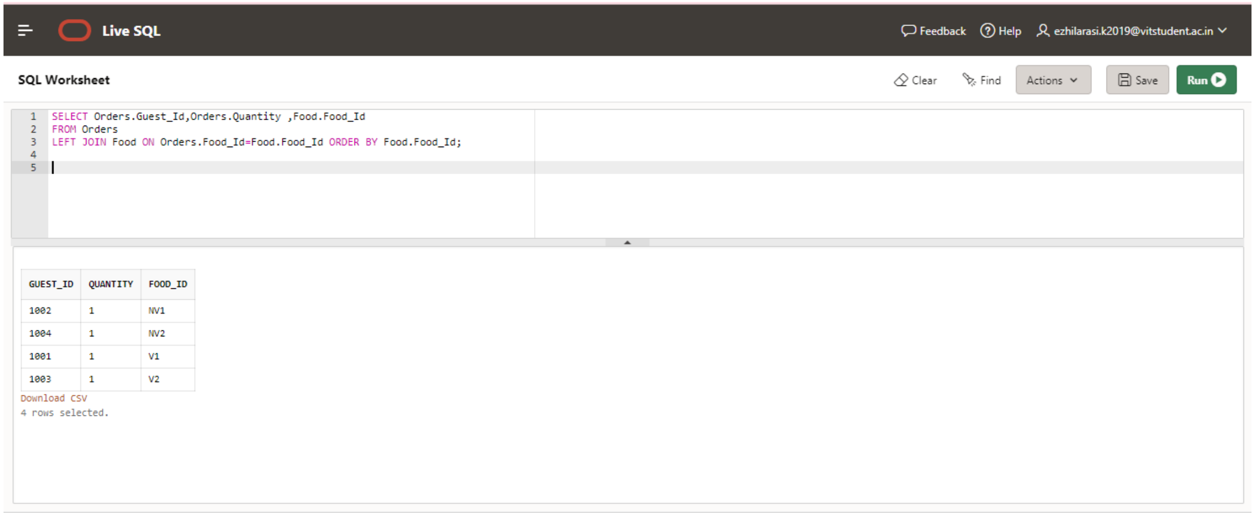
JOIN :

Left join:

SELECT Orders.Guest_Id,Orders.Quantity

,Food.Food_IdFROM Orders

LEFT JOIN Food ON Orders.Food_Id=Food.Food_Id ORDER BY Food.Food_Id;



```
SELECT Orders.Guest_Id,Orders.Quantity ,Food.Food_Id
FROM Orders
LEFT JOIN Food ON Orders.Food_Id=Food.Food_Id ORDER BY Food.Food_Id;
```

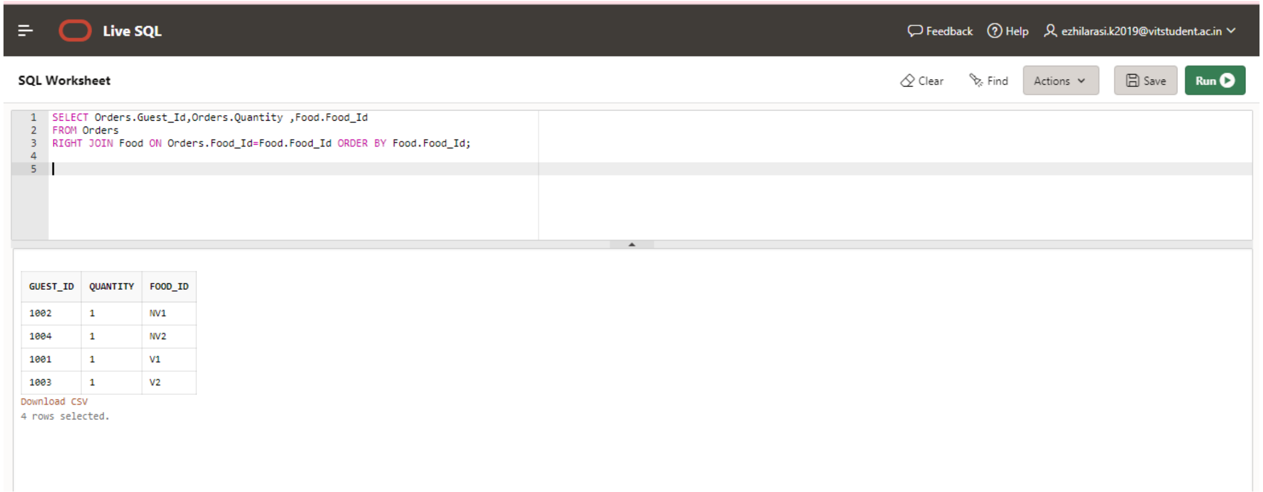
GUEST_ID	QUANTITY	FOOD_ID
1002	1	NV1
1004	1	NV2
1001	1	V1
1003	1	V2

Download CSV
4 rows selected.

SELECT Orders.Guest_Id,Orders.Quantity

,Food.Food_IdFROM Orders

RIGHT JOIN Food ON Orders.Food_Id=Food.Food_Id ORDER BY Food.Food_Id;



```
SELECT Orders.Guest_Id,Orders.Quantity ,Food.Food_Id
FROM Orders
RIGHT JOIN Food ON Orders.Food_Id=Food.Food_Id ORDER BY Food.Food_Id;
```

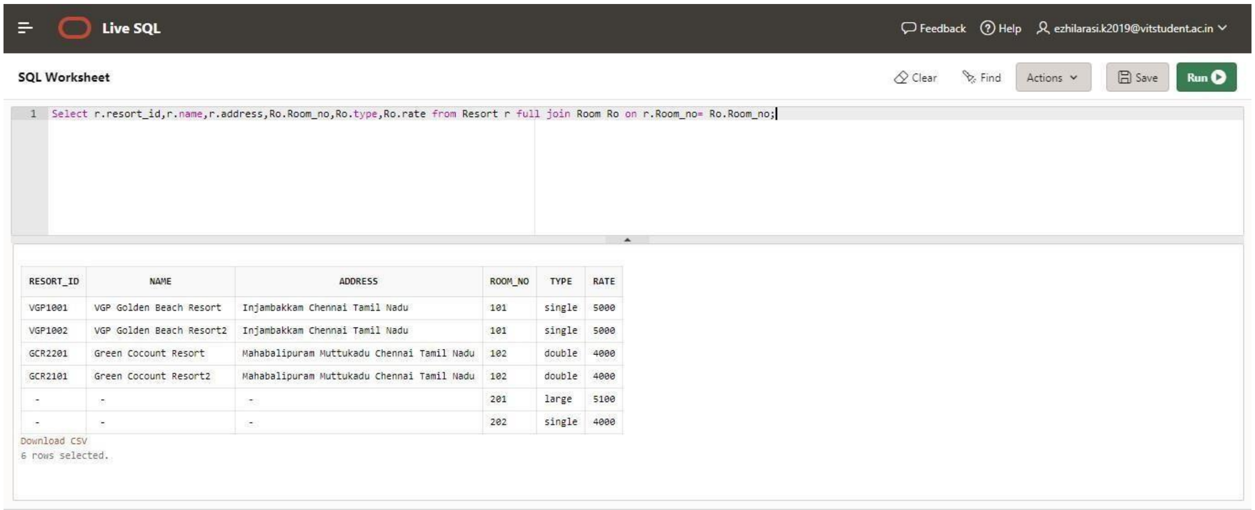
GUEST_ID	QUANTITY	FOOD_ID
1002	1	NV1
1004	1	NV2
1001	1	V1
1003	1	V2

Download CSV
4 rows selected.

Full join:

display the room and resort details(Using left/right/full outer join query).

Select r.resort_id,r.name,r.address,Ro.Room_no,Ro.type,Ro.rate from Resortr full join Room Ro on r.Room_no= Ro.Room_no;



```sql
Select r.resort_id,r.name,r.address,Ro.Room_no,Ro.type,Ro.rate from Resort r full join Room Ro on r.Room_no= Ro.Room_no;
```

RESORT_ID	NAME	ADDRESS	ROOM_NO	TYPE	RATE
VGP1001	VGP Golden Beach Resort	Injambakkam Chennai Tamil Nadu	101	single	5000
VGP1002	VGP Golden Beach Resort2	Injambakkam Chennai Tamil Nadu	101	single	5000
GCR2201	Green Cocount Resort	Mahabalipuram Muttukadu Chennai Tamil Nadu	102	double	4000
GCR2101	Green Cocount Resort2	Mahabalipuram Muttukadu Chennai Tamil Nadu	102	double	4000
-	-	-	201	large	5100
-	-	-	202	single	4000

Download CSV
6 rows selected.

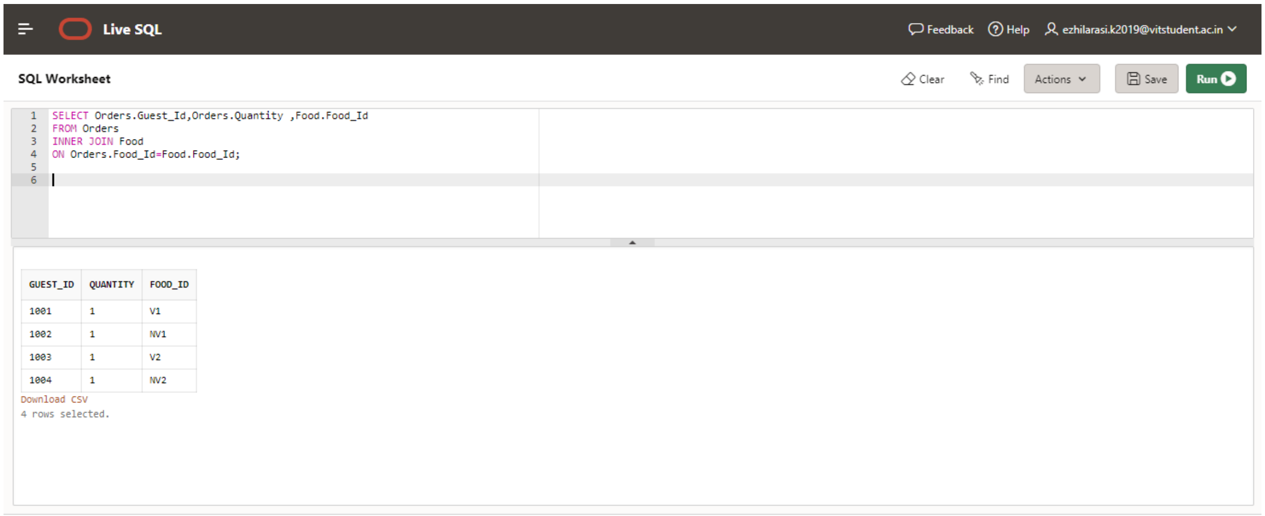
Inner join:

SELECT

Orders.Guest_Id,Orders.Quantity,Food.Food_Id

FROM Orders INNER JOIN Food ON

Orders.Food_Id=Food.Food_Id;



```sql
SELECT Orders.Guest_Id,Orders.Quantity ,Food.Food_Id
FROM Orders
INNER JOIN Food
ON Orders.Food_Id=Food.Food_Id;
```

GUEST_ID	QUANTITY	FOOD_ID
1001	1	V1
1002	1	NV1
1003	1	V2
1004	1	NV2

Download CSV
4 rows selected.

7.Query Optimization

1)Relation:

Room({RoomNo,Rate,Single,Double,Large})

Resort({Rname,ResortId,address,RoomNum})

Workers({Wname,wrkId,address,ResId})

SQL Query:

SELECT Rname

FROM Room,Resort,Workers

WHERE Wname='Priya' AND ResortId=ResId AND RoomNO=RoomNum AND Rate<'20000';

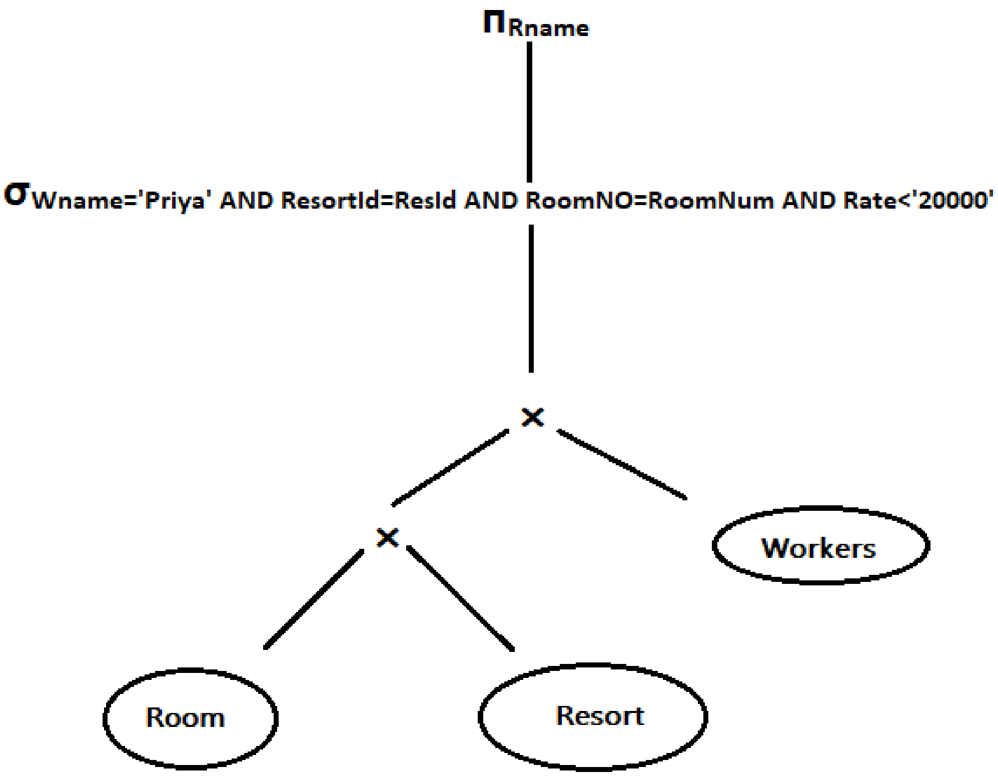
Relational Algebra:

$$\pi_{\text{Rname}}\ (\sigma_{\text{Wname='Priya' AND ResortId=ResId AND RoomNO=RoomNum AND Rate<'20000'}}(\text{Workers} \times (\text{Resort} \times \text{Room})))$$

QUERY TREE:

STEP1:

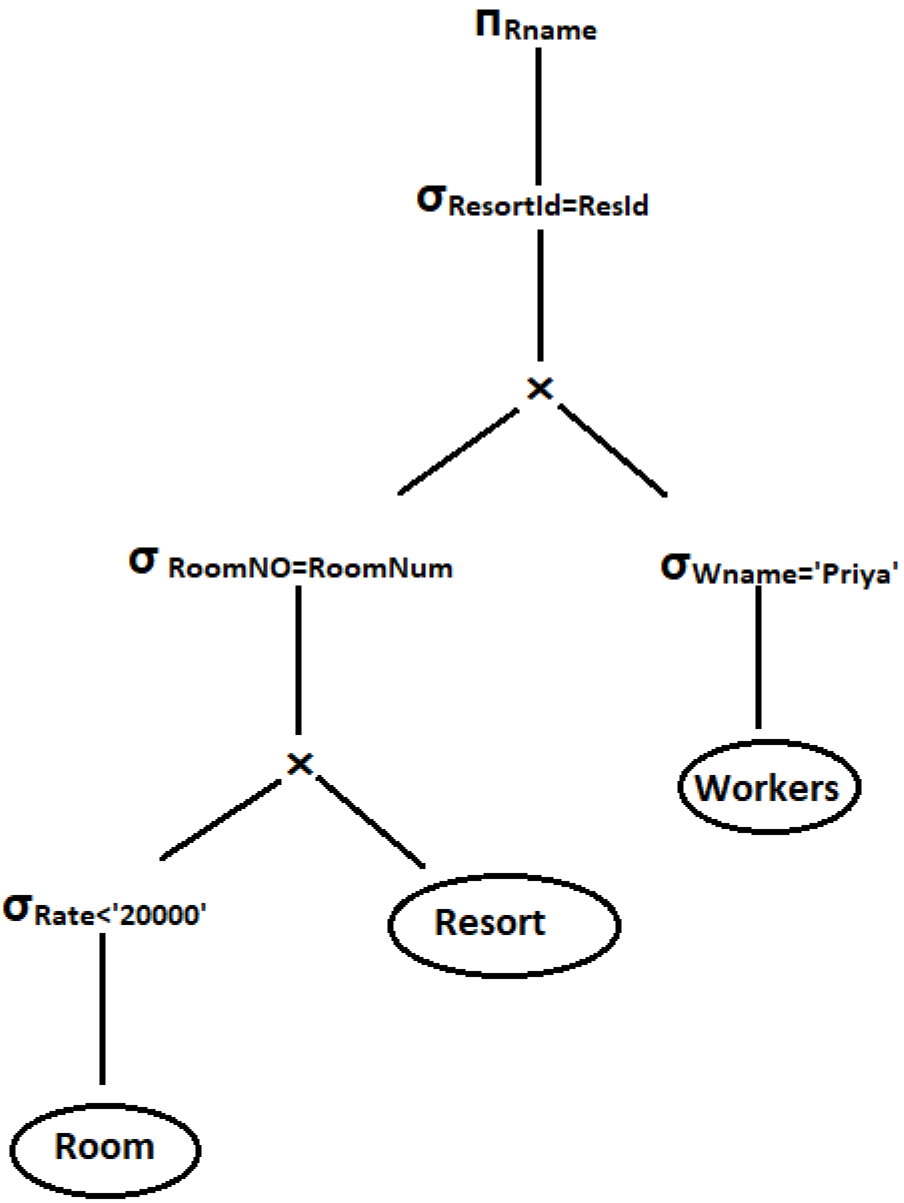
Initial (canonical) Query tree for SQL query.


πRname
σWname='Priya' AND ResortId=ResId AND RoomNO=RoomNum AND Rate<'20000'
×
×
Workers
Room
Resort

STEP2:

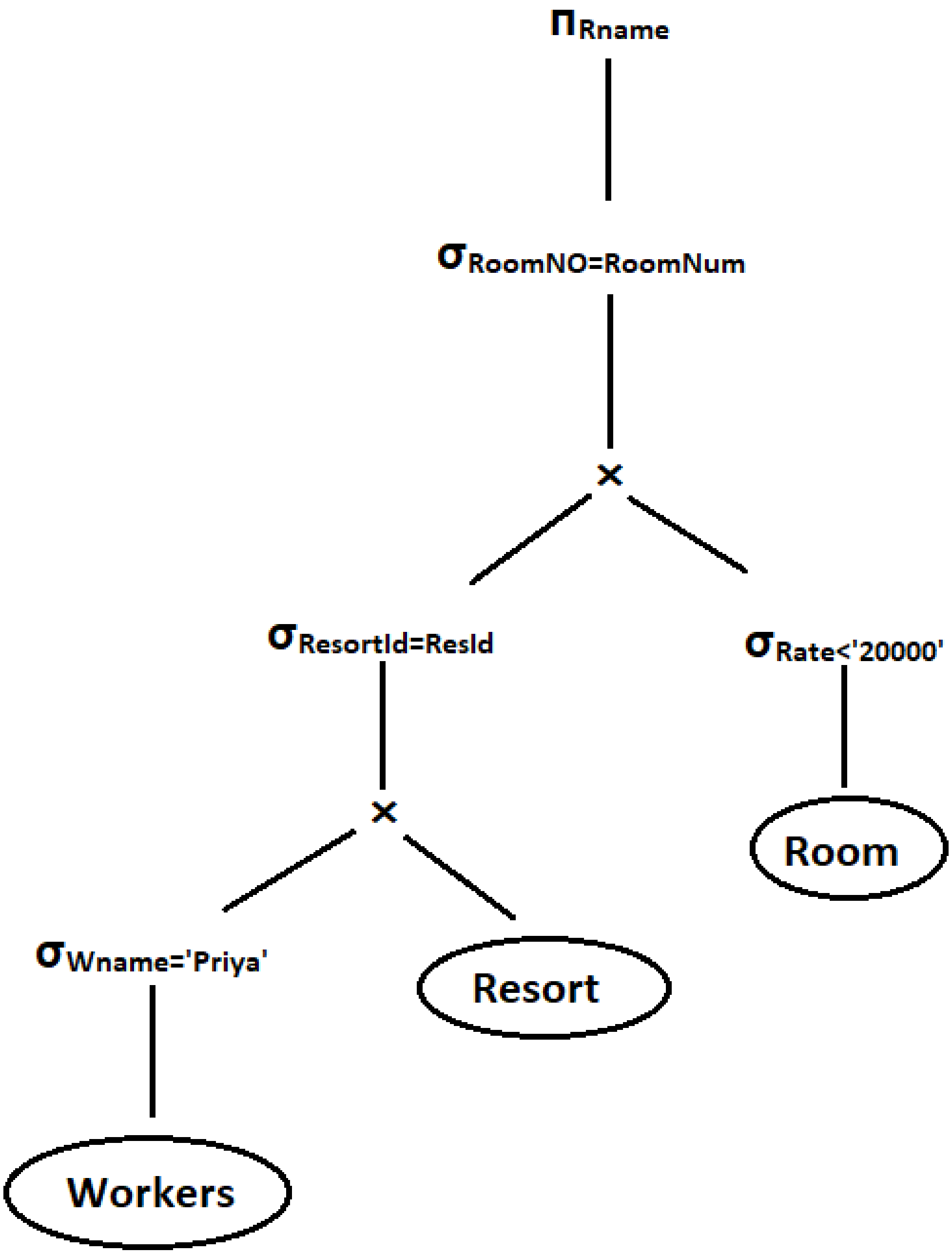
Moving SELECT operations down the Query tree.

20BIT0309 / 20BIT0303/19BIT0395



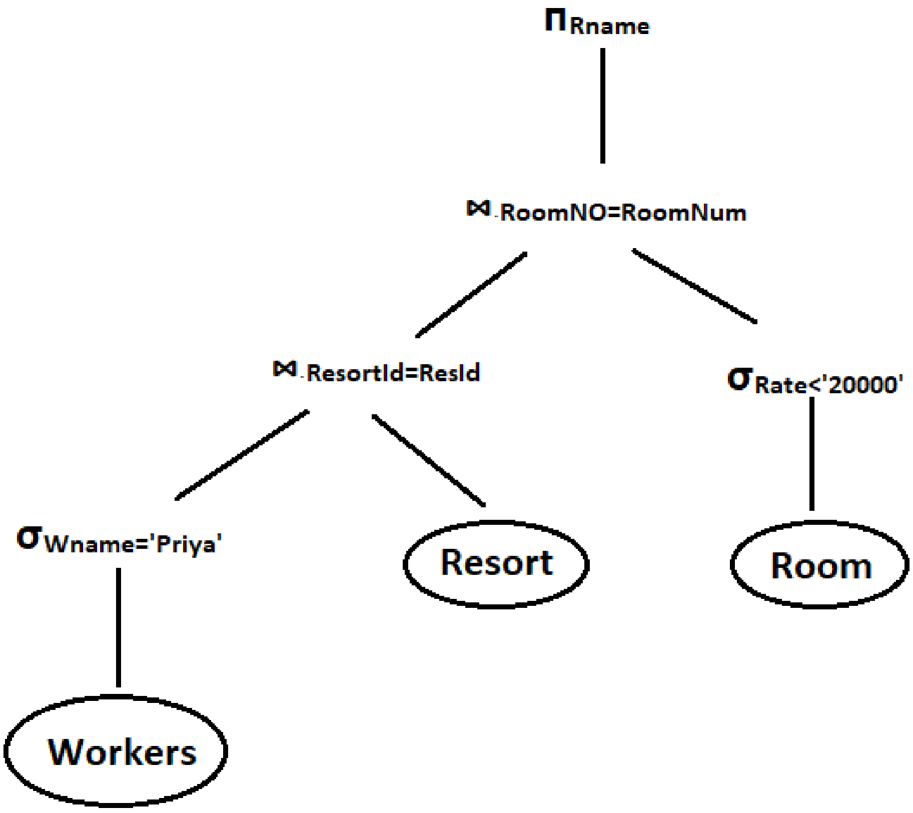
STEP3:

Applying The move restrictive SELECT operation first.



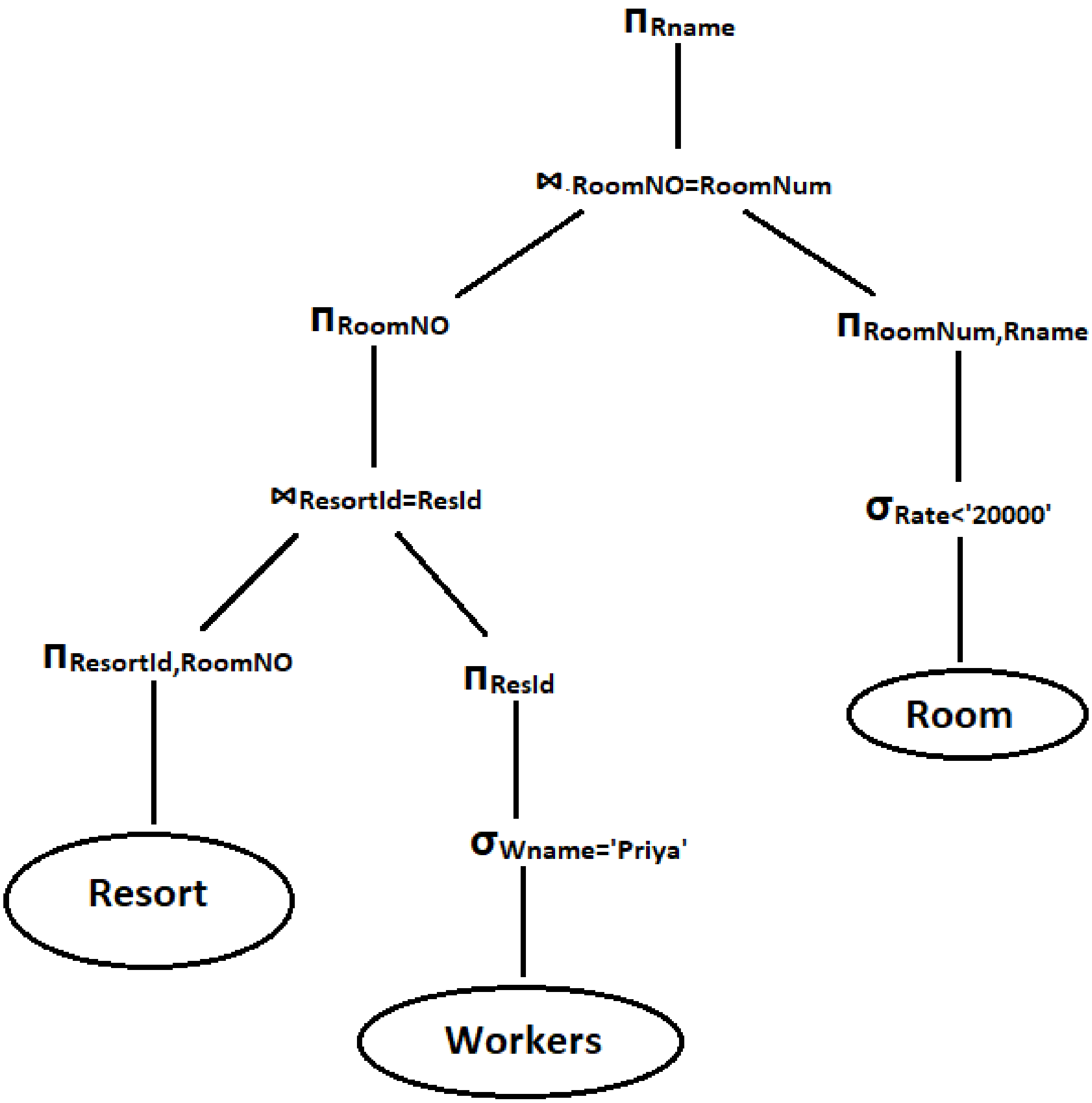
STEP4:

Replacing CARTESIAN product and SELECT with JOIN operation.

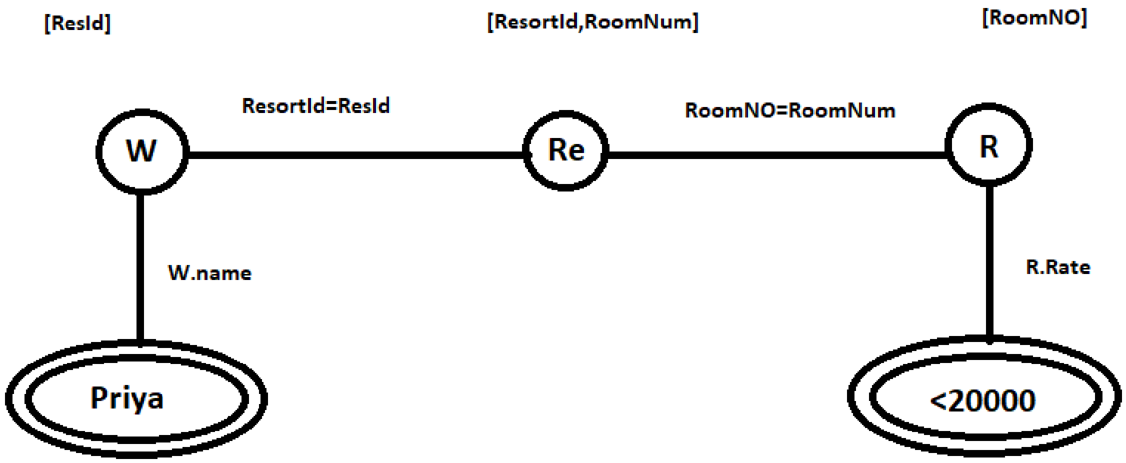


STEP5:

Moving PROJECT operations down the Query tree.



Query Graph:



2)Relation:

Food({FoodId,Veg,Non-Veg,Rate})

Orders({Time,Quantity,Date,GuestId,FoodId})

Guest({Name,GuestId,Address,RoomNo,CheckOutDate,CheckInDate}).

SQL Query:

SELECT GuestId,Name

FROM Food F, Orders O, Guest G

WHERE F.FoodId=O.FoodId AND O.GuestId=G.GuestId AND RoomNO='101' AND Date>'21-sep-2020' AND FoodId='200';

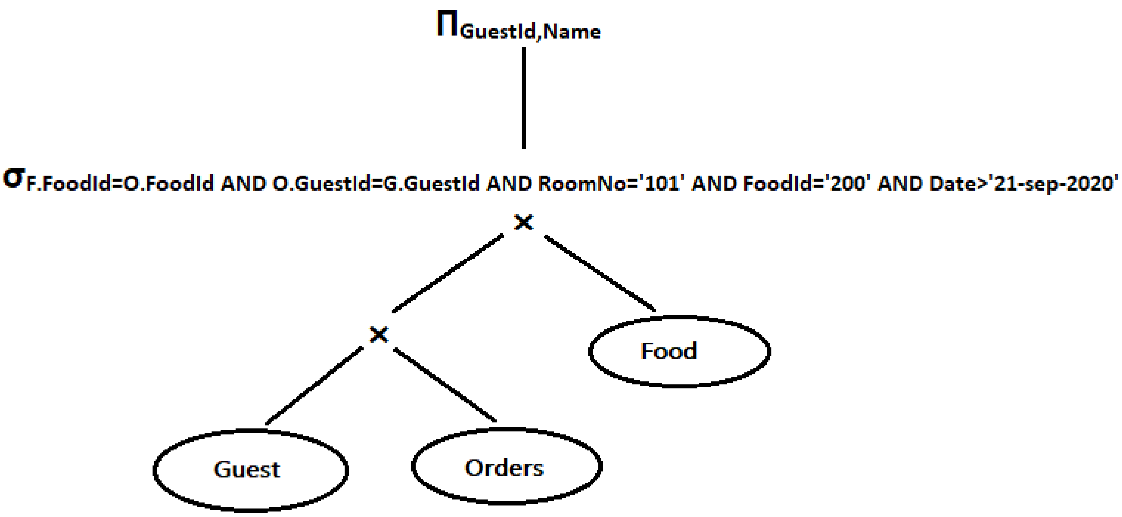
Relational Algebra:

$\Pi_{\text{GuestId,Name}}\ (\sigma_{\text{F.FoodId=O.FoodId AND O.GuestId=G.GuestId AND RoomNo='101' AND FoodId='200' AND Date>'21-sep-2020'}}\ (\text{Food} \times (\text{Guest} \times \text{Orders})))$

QUERY TREE:

STEP1:

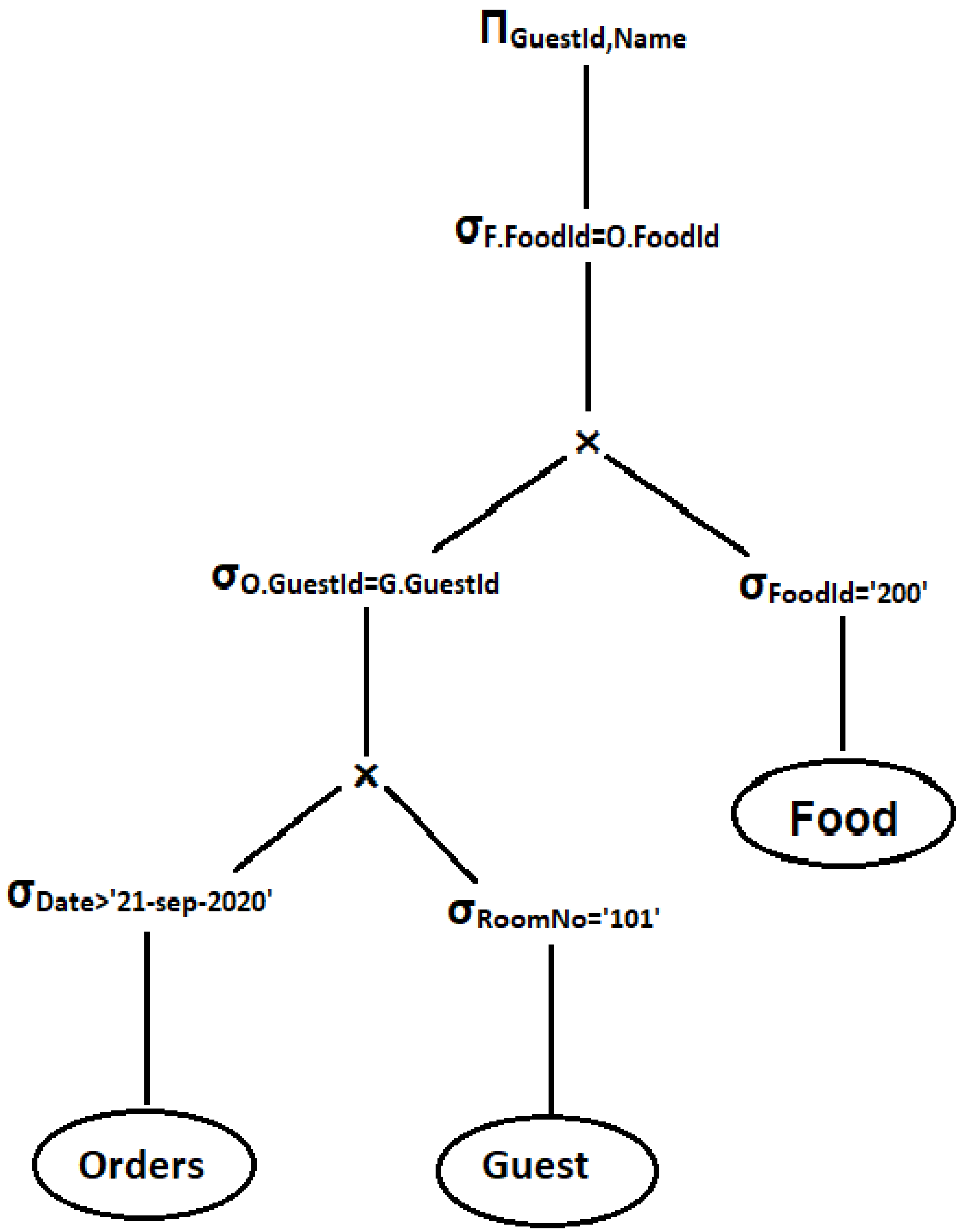
Initial (canonical) Query tree for SQL query.



$\Pi_{\text{GuestId,Name}}$
$\sigma_{\text{F.FoodId=O.FoodId AND O.GuestId=G.GuestId AND RoomNo='101' AND FoodId='200' AND Date>'21-sep-2020'}}$
×
×
Food
Guest
Orders

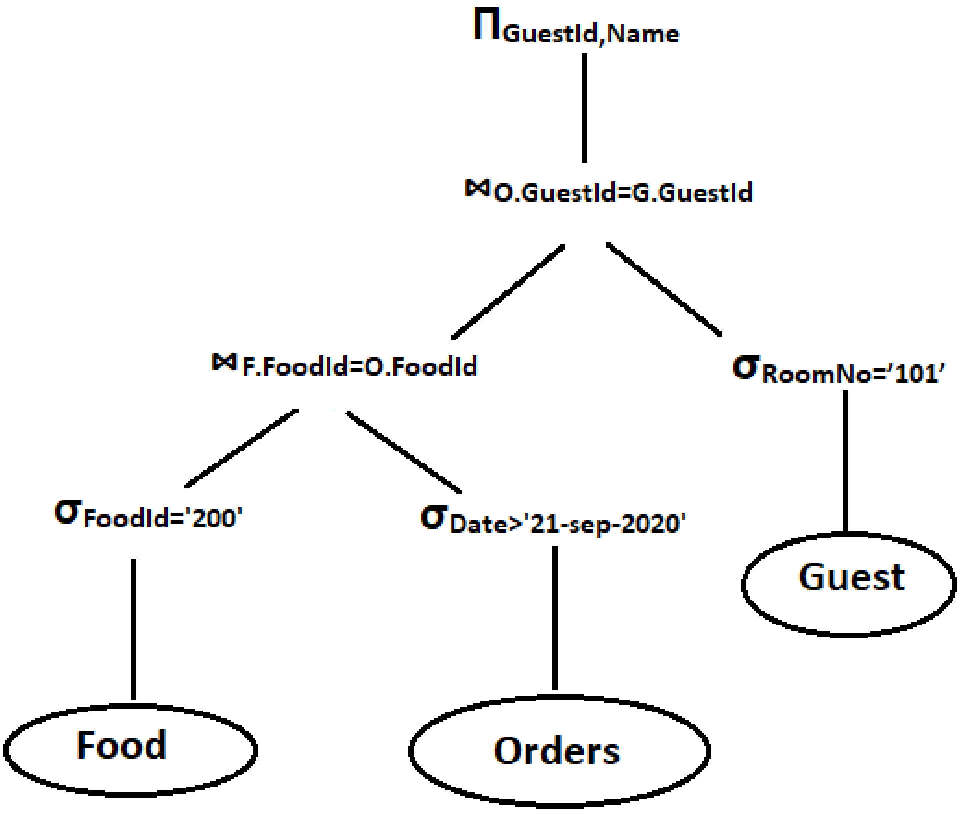
STEP2:

Moving SELECT operations down the Query tree.



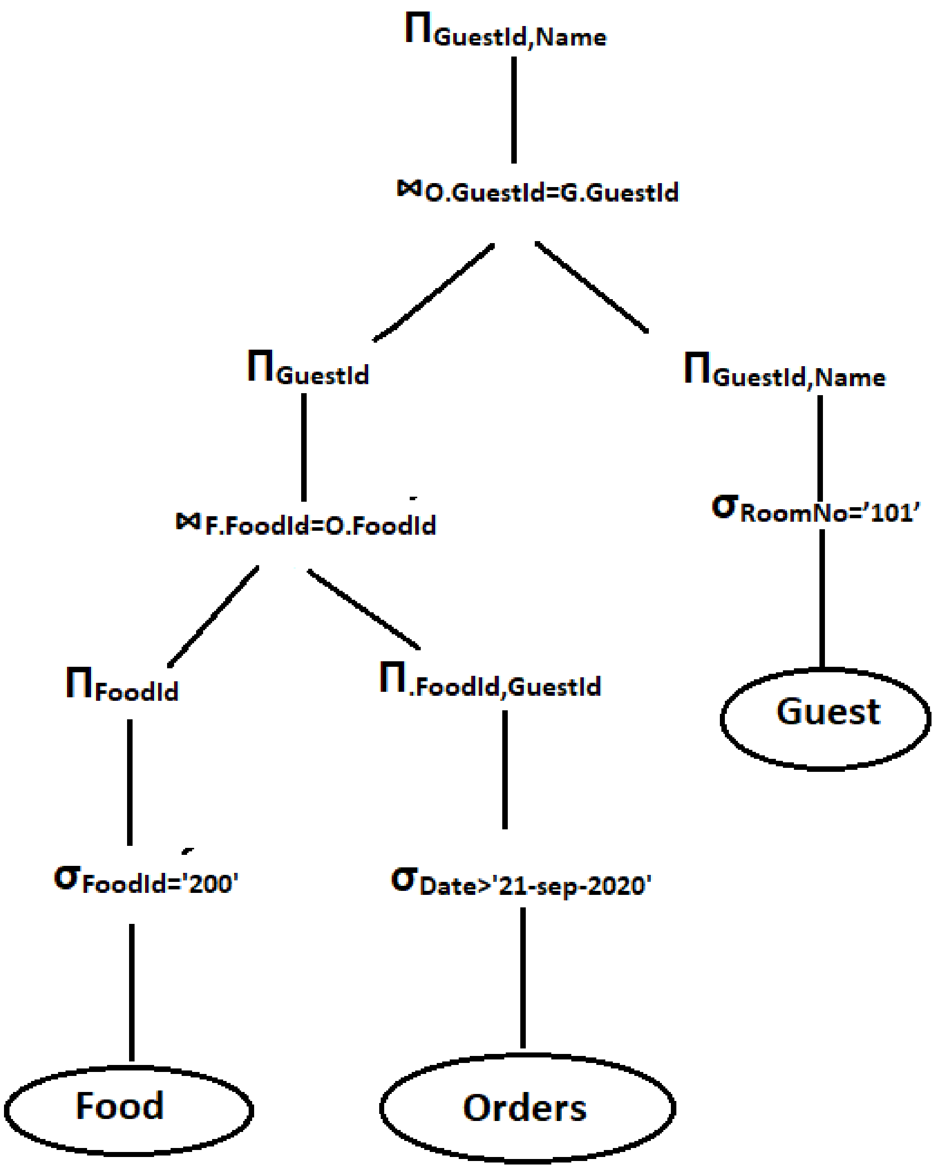
STEP4:

Replacing CARTESIAN product and SELECT with JOIN operation.

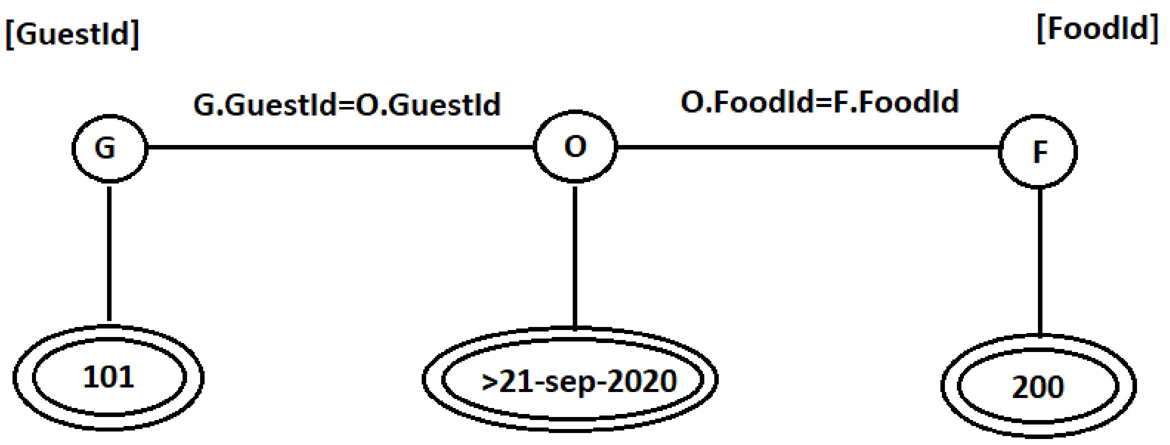


STEP5:

Moving PROJECT operations down the Query tree.



Query Graph:



8.Indexing

Definition

Indexing is a way to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed. It is a data structure technique which is used to quickly locate and access the data in a database. Indexes are created using a few database columns.

Index can be represented as

Search Key	Pointer

- search key that contains a copy of the primary key or candidate key of the table.
- Pointer contains a set of pointers holding the address of the disk block where the value of the particular key can be found.

Indexing Methods:

Primary Index

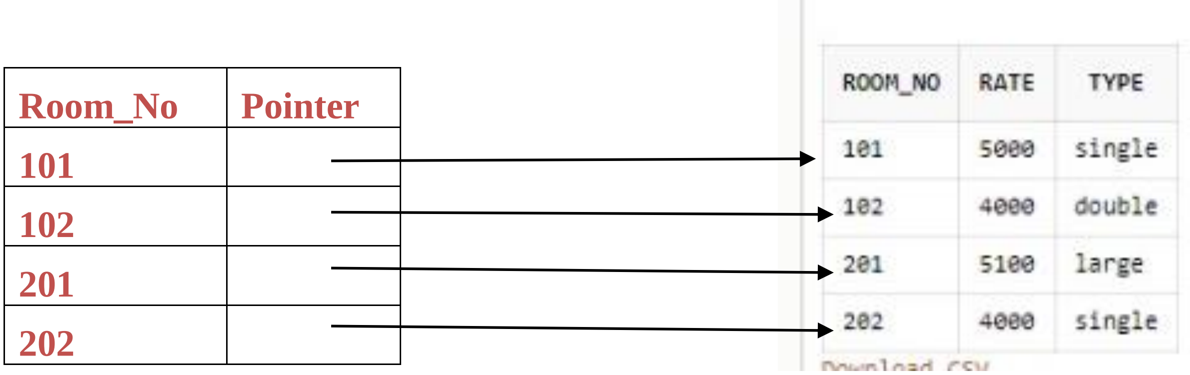
- If the index is created on the basis of the primary key of the table, then it is known as primary indexing. These primary keys are unique to each record and contain 1:1 relation between the records.
- As primary keys are stored in sorted order, the performance of the searching operation is quite efficient.
- The primary index can be classified into two types: Dense index and Sparse index.

Dense index

- The dense index contains an index record for every search key value in the data file. It makes searching faster.
- **In this, the number of records in the index table is same as the number of records in the main table.**
- It needs more space to store index record itself. The index records have the search key and a pointer to the actual record on the disk.

In our table

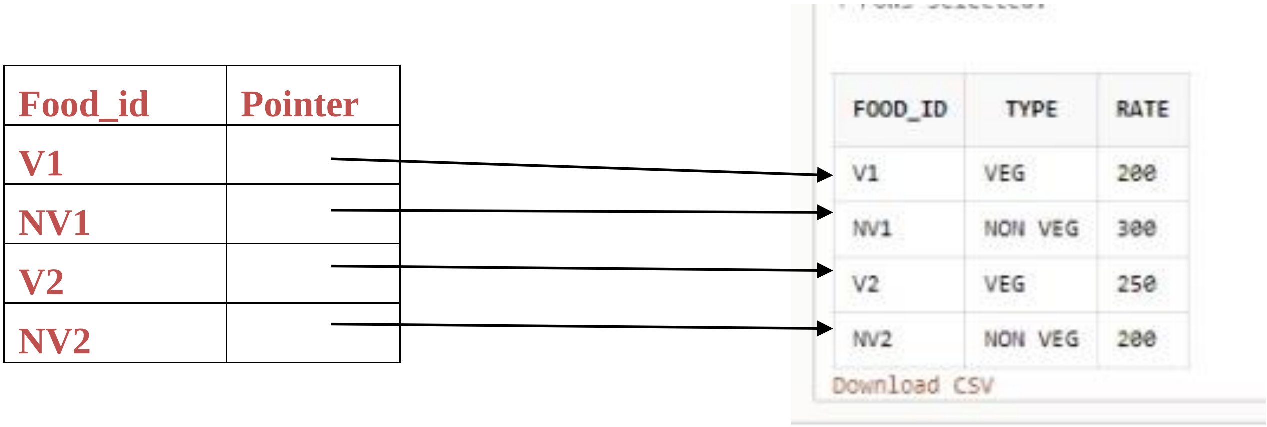
Room



Room_No	Pointer
101	
102	
201	
202	

ROOM_NO	RATE	TYPE
101	5000	single
102	4000	double
201	5100	large
202	4000	single

Food



Food_id	Pointer
V1	
NV1	
V2	
NV2	

FOOD_ID	TYPE	RATE
V1	VEG	200
NV1	NON VEG	300
V2	VEG	250
NV2	NON VEG	200

Dense Indexing can be applied to any of the tables since primary key is unsorted .

Sparse index

- In the data file, index record appears only for a few items. Each item points to a block.
- In this, instead of pointing to each record in the main table, the index points to the records in the main table in a gap.

In our table

Sparse Indexing cannot be applied to any of the tables since primary key is unsorted .

9.Conclution

One of the most fundamental objectives of an online resort reservation system is to increase the global reach of a resort website. Through it, resort managers are able to launch resort website so that various users around the world can access the website. Thus it increases the chances of a number of bookings. Also, it saves a lot of time and effort for customers as they can conveniently book resort from their homes. "Resort management" to put it simply, means "Booking of Resort ,Resort room, Hall Order etc. also provide a membership, Club Activities options to customer. Using this system we can find out room availability, booking date and time, check in and check out time and many more options, so it will take small time to find out all this things. Resort management keeps all information of customer membership details using membership id so when customer comes in resort and provide membership id to admin then he check quickly id is valid or not. Also assign club activity selected by customer to membership id. Billing options is available in this system. The system is user-friendly means it is easy to use. Resort management keeps Visited Customer details, Employee details in to database. Database use in this system is helpful to find out performance of resort, sales per year, and other information. Databases in Resort management are very dynamic and scalable.

Thankyou.

…………………………………………..