EXPERIMENT 1& 2:

Link:

Experiment 1:

https://www.youtube.com/watch?v=gGLnmHdKCW0

Experiment 2:

https://www.youtube.com/watch?v=NS46q3yeVCU

DESCRIPTION:

An SQL relation is defined using thecreate tablecommand:

create table r (A_1D_1, A_2D_2, ..., A_nD_n,
(integrity-constraint$_1$),
...,
(integrity-constraint$_k$))

r is the name of the relation

eachA_i is an attribute name in the schema of relation r

D_i is the data type of values in the domain of attribute A_i

Unique Key

The unique constraint doesn't allow duplicate values in a column. If the unique constraint encompasses two or more columns, no two equal combinations are allowed.

Check

Check constraint is used to restrict the values before inserting into a table.

Primary Key

The key column with which we can identify the entire Table is called as a primary key column.

A primary key is a combination of a Unique and a Not Null constraint, it will not allow null and duplicate values.

A table can have only one primary key. A primary key can be declared on two or more columns as a Composite Primary Key.

ALTER TABLE

Syntax

Adding a column to table:

Alter table <table name > add <new column name><type>

Modifying the column with new datatype:

Alter table <table name>modify<column name><new type>

Dropping a column:

Alter table<table name> drop column <column name>

Renaming column Name:

Alter table<table name> rename column <column name> to < new column name>

Rename Table:

Rename <tablename>to <new_tablename>

Add Constraint to Table:

ALTER TABLE <table name> ADD CONSTRAINT <constraint name><constraint type>

(column name);

Alter table <table name><modify ><column name><constraint type>

ALTER TABLE <table name> DROP CONSTRAINT <constraint name>

CODE:

EXPERIMENT 1:

SQL> create table Course_ur14cs290(coursecode number(2) not null,coursename varchar2(30),syllabus varchar2(60));

Table created.

SQL> create table Batch_ur14cs290(batchcode number(3) not null, coursecode number(2) not null,startingdate date not null,duration number(3) not null, coursefees number(10,2) not null);

Table created.

SQL> create table Enquiry_ur14cs290(enquiryno number(8) not null,fname varchar2(30),sname varchar2(30), coursecode number(2),street varchar2(30),city varchar2(30),pincode number(6), phone number(10),enquirydate date not null,en_st char(1));

Table created.

SQL> create table Enrollment_ur14cs290(rollno number(8) not null, enquiryno number(8), batchcode number(5), enrollmentdate date not null);

Table created.

SQL> create table Feespaid_ur14cs290(rollno number(8) not null,feespaiddate date not null,chequeno varchar2(10),bankname varchar2(30),amount number(10,2));

Table created.

SQL> insert into Batch_ur14cs290 values (&batchcode,&coursecode,'&startingdate',&duration,&coursefess);

Enter value for batchcode: 101

Enter value for coursecode: 20

Enter value for startingdate: 01-Jan-14

Enter value for duration: 30

Enter value for coursefess: 5000

old 1: insert into Batch_ur14cs290 values (&batchcode,&coursecode,'&startingdate',&duration,&coursefess)

new 1: insert into Batch_ur14cs290 values (101,20,'01-Jan-14',30,5000)

1 row created.

SQL> /

Enter value for batchcode: 102

Enter value for coursecode: 10

Enter value for startingdate: 20-Jan-14

Enter value for duration: 20

Enter value for coursefess: 4000

old 1: insert into Batch_ur14cs290 values (&batchcode,&coursecode,'&startingdate',&duration,&coursefess)

new 1: insert into Batch_ur14cs290 values (102,10,'20-Jan-14',20,4000)

1 row created.

SQL> /

Enter value for batchcode: 103

Enter value for coursecode: 10

Enter value for startingdate: 01-Apr-14

Enter value for duration: 20

Enter value for coursefess: 4000

old 1: insert into Batch_ur14cs290 values
(&batchcode,&coursecode,'&startingdate',&duration,&coursefess)

new 1: insert into Batch_ur14cs290 values (103,10,'01-Apr-14',20,4000)

1 row created.

SQL> /

Enter value for batchcode: 104

Enter value for coursecode: 10

Enter value for startingdate: 25-Oct-14

Enter value for duration: 20

Enter value for coursefess: 4000

old 1: insert into Batch_ur14cs290 values
(&batchcode,&coursecode,'&startingdate',&duration,&coursefess)

new 1: insert into Batch_ur14cs290 values (104,10,'25-Oct-14',20,4000)

1 row created.

SQL> insert into Enquiry_ur14cs290 values
(&eno,'&fname','&sname',&coursecode,'&street','&city',&pincode,&phone,'&enquiry','&en');

Enter value for eno: 110001

Enter value for fname: Anil

Enter value for sname: Sharma

Enter value for coursecode: 10

Enter value for street: Ramnagar

Enter value for city: Cbe

Enter value for pincode: 641114

Enter value for phone: 9486453331

Enter value for enquiry: 20-Jan-14

Enter value for en: Y

old 1: insert into Enquiry_ur14cs290 values
(&eno,'&fname','&sname',&coursecode,'&street','&city',&pincode,&phone,'&enquiry','&en')

new 1: insert into Enquiry_ur14cs290 values
(110001,'Anil','Sharma',10,'Ramnagar','Cbe',641114,9486453331,'20-Jan-14','Y')

1 row created.

SQL> /

Enter value for eno: 110002

Enter value for fname: Sachin

Enter value for sname: Nimje

Enter value for coursecode: 20

Enter value for street: Gandhi nagar

Enter value for city: ERD

Enter value for pincode: 641142

Enter value for phone: 9234567890

Enter value for enquiry: 25-Oct-14

Enter value for en: N

old 1: insert into Enquiry_ur14cs290 values
(&eno,'&fname','&sname',&coursecode,'&street','&city',&pincode,&phone,'&enquiry','&en')

new 1: insert into Enquiry_ur14cs290 values (110002,'Sachin','Nimje',20,'Gandhi
nagar','ERD',641142,9234567890,'25-Oct-14','N')

1 row created.

SQL> insert into Enrollment_ur14cs290 values (&rno,&eno,&batchcode,'&edate');

Enter value for rno: 20002001

Enter value for eno: 110001

Enter value for batchcode: 101

Enter value for edate: 01-Jul-13

old 1: insert into Enrollment_ur14cs290 values (&rno,&eno,&batchcode,'&edate')

new 1: insert into Enrollment_ur14cs290 values (20002001,110001,101,'01-Jul-13')

1 row created.

SQL> /

Enter value for rno: 20002002

Enter value for eno: 110002

Enter value for batchcode: 102

Enter value for edate: 25-Oct-13

old 1: insert into Enrollment_ur14cs290 values (&rno,&eno,&batchcode,'&edate')

new 1: insert into Enrollment_ur14cs290 values (20002002,110002,102,'25-Oct-13')

1 row created.

SQL> /

Enter value for rno: 20002003

Enter value for eno: 110021

Enter value for batchcode: 103

Enter value for edate: 11-Jan-14

old 1: insert into Enrollment_ur14cs290 values (&rno,&eno,&batchcode,'&edate')

new 1: insert into Enrollment_ur14cs290 values (20002003,110021,103,'11-Jan-14')

1 row created.

SQL> /

Enter value for rno: 20002004

Enter value for eno: 110022

Enter value for batchcode: 104

Enter value for edate: 29-Jan-14

old 1: insert into Enrollment_ur14cs290 values (&rno,&eno,&batchcode,'&edate')

new 1: insert into Enrollment_ur14cs290 values (20002004,110022,104,'29-Jan-14')

1 row created.

SQL> /

Enter value for rno: 20002005

Enter value for eno: 110023

Enter value for batchcode: 110

Enter value for edate: 01-Mar-14

old 1: insert into Enrollment_ur14cs290 values (&rno,&eno,&batchcode,'&edate')

new 1: insert into Enrollment_ur14cs290 values (20002005,110023,110,'01-Mar-14')

1 row created.

SQL> insert into Feespaid_ur14cs290 values
(&rollno,'&feespaiddate','&chequeno','&bankname',&amount);

Enter value for rollno: 20002001

Enter value for feespaiddate: 01-Jul-133

Enter value for chequeno: 123456

Enter value for bankname: HDFC

Enter value for amount: 5000

old 1: insert into Feespaid_ur14cs290 values
(&rollno,'&feespaiddate','&chequeno','&bankname',&amount)

new 1: insert into Feespaid_ur14cs290 values (20002001,'01-Jul-133','123456','HDFC',5000)

1 row created.

SQL> /

Enter value for rollno: 20002002

Enter value for feespaiddate: 25-Oct-13

Enter value for chequeno: -

Enter value for bankname: -

Enter value for amount: 4000

old 1: insert into Feespaid_ur14cs290 values
(&rollno,'&feespaiddate','&chequeno','&bankname',&amount)

new 1: insert into Feespaid_ur14cs290 values (20002002,'25-Oct-13','-','-',4000)

1 row created.

SQL> /

Enter value for rollno: 20002003

Enter value for feespaiddate: 11-Jan-14

Enter value for chequeno: 876544

Enter value for bankname: TMB

Enter value for amount: 3000

old 1: insert into Feespaid_ur14cs290 values
(&rollno,'&feespaiddate','&chequeno','&bankname',&amount)

new 1: insert into Feespaid_ur14cs290 values (20002003,'11-Jan-14','876544','TMB',3000)

1 row created.

SQL> /

Enter value for rollno: 20002004

Enter value for feespaiddate: 29-Jan-14

Enter value for chequeno: 765433

Enter value for bankname: DBI

Enter value for amount: 4000

old 1: insert into Feespaid_ur14cs290 values (&rollno,'&feespaiddate','&chequeno','&bankname',&amount)

new 1: insert into Feespaid_ur14cs290 values (20002004,'29-Jan-14','765433','DBI',4000)

1 row created.

SQL> /

Enter value for rollno: 20002005

Enter value for feespaiddate: 01-Mar-14

Enter value for chequeno: -

Enter value for bankname: -

Enter value for amount: 5000

old 1: insert into Feespaid_ur14cs290 values (&rollno,'&feespaiddate','&chequeno','&bankname',&amount)

new 1: insert into Feespaid_ur14cs290 values (20002005,'01-Mar-14','-','-',5000)

1 row created.

SQL> insert into Course_ur14cs290 values (&coursecode,'&coursename','&syllabus');

Enter value for coursecode: 10

Enter value for coursename: Oracle

Enter value for syllabus: Sql, plsql, jdbc,jsp

old 1: insert into Course_ur14cs290 values (&coursecode,'&coursename','&syllabus')

new 1: insert into Course_ur14cs290 values (10,'Oracle','Sql, plsql, jdbc,jsp')

```
1 row created.

SQL> /
Enter value for coursecode: 20
Enter value for coursename: Java
Enter value for syllabus: Core java,servelets,jsp,ejb
old   1: insert into Course_ur14cs290 values (&coursecode,'&coursename','&syllabus')
new   1: insert into Course_ur14cs290 values (20,'Java','Core java,servelets,jsp,ejb')

1 row created.

SQL> select * from Course_ur14cs290;

COURSECODE COURSENAME                SYLLABUS
---------- ------------------------- ------------------------------------------------------------
        10 Oracle                    Sql, plsql, jdbc,jsp
        20 Java                      Core java,servelets,jsp,ejb

SQL> select * from Batch_ur14cs290;

 BATCHCODE COURSECODE STARTINGD   DURATION COURSEFEES
---------- ---------- --------- ---------- ----------
       101         20 01-JAN-14         30       5000
       102         10 20-JAN-14         20       4000
       103         10 01-APR-14         20       4000
```

104 10 25-OCT-14 20 4000

SQL> select * from Enquiry_ur14cs290;

ENQUIRYNO FNAME SNAME COURSECODE STREET
CITY PINCODE PHONE ENQUIRYDA E

---------- ---------- ---------- ---------- ---------- ---
---------- ---------- ---------- --------- -

110001 Anil Sharma 10 Ramnagar Cbe
641114 9486453331 20-JAN-14 Y

110002 Sachin Nimje 20 Gandhi nagar ERD
641142 9234567890 25-OCT-14 N

SQL> select * from Enrollment_ur14cs290;

ROLLNO ENQUIRYNO BATCHCODE ENROLLMEN

---------- ---------- ---------- ---------

20002001 110001 101 01-JUL-13

20002002 110002 102 25-OCT-13

20002003 110021 103 11-JAN-14

20002004 110022 104 29-JAN-14

20002005 110023 110 01-MAR-14

SQL> select * from Feespaid_ur14cs290;

ROLLNO FEESPAIDD CHEQUENO BANKNAME AMOUNT

---------- --------- ---------- ---------- ----------

20002001 01-JUL-33 123456 HDFC 5000

20002002 25-OCT-13 - - 4000

20002003 11-JAN-14 876544 TMB 3000

20002004 29-JAN-14 765433 DBI 4000

20002005 01-MAR-14 - - 5000

SQL> commit;

Commit complete.

SQL> spool off;

EXPERIMENT 2:

SQL> desc Course_ur14cs290;

Name	Null?	Type
COURSECODE	NOT NULL	NUMBER(2)
COURSENAME		VARCHAR2(30)
SYLLABUS		VARCHAR2(60)

SQL> desc Batch_ur14cs290;

Name	Null?	Type
BATCHCODE	NOT NULL	NUMBER(3)

COURSECODE	NOT NULL	NUMBER(2)
STARTINGDATE	NOT NULL	DATE
DURATION	NOT NULL	NUMBER(3)s
COURSEFEES	NOT NULL	NUMBER(10,2)

SQL> desc Enquiry_ur14cs290;

Name	Null?	Type
ENQUIRYNO	NOT NULL	NUMBER(8)
FNAME		VARCHAR2(30)
SNAME		VARCHAR2(30)
COURSECODE		NUMBER(2)
STREET		VARCHAR2(30)
CITY		VARCHAR2(30)
PINCODE		NUMBER(6)
PHONE		NUMBER(10)
ENQUIRYDATE	NOT NULL	DATE
EN_ST		CHAR(1)

SQL> desc Enrollment_ur14cs290;

Name	Null?	Type
ROLLNO	NOT NULL	NUMBER(8)
ENQUIRYNO	NOT NULL	NUMBER(8)
BATCHCODE		NUMBER(5)

ENROLLMENTDATE NOT NULL DATE

SQL> desc Feespaid_ur14cs290;

Name	Null?	Type
ROLLNO	NOT NULL	NUMBER(8)
FEESPAIDDATE	NOT NULL	DATE
CHEQUENO		VARCHAR2(10)
BANKNAME		VARCHAR2(30)
AMOUNT		NUMBER(10,2)

SQL> alter table Enrollment_ur14cs290 add constraint p1 primary key(enquiryno);

Table altered.

SQL> truncate table Enrollment_ur14cs290;

Table truncated.

SQL> alter table Enquiry_ur14cs290 add constraint f1 foreign key (enquiryno) references Enrollment_ur14cs290 (enquiryno);

Table altered.

SQL> alter table Course_ur14cs290 add constraint p2 primary key(coursecode);

Table altered.

SQL> truncate table Batch_ur14cs290;

Table truncated.

SQL> alter table Batch_ur14cs290 modify batchcode varchar(20);

Table altered.

SQL> alter table Enrollment_ur14cs290 add constraint unr1 unique (rollno);

Table altered.

SQL> alter table Enquiry_ur14cs290 drop column en_st;

Table altered.

SQL> desc Enquiry_ur14cs290;

Name	Null?	Type
ENQUIRYNO	NOT NULL	NUMBER(8)
FNAME		VARCHAR2(30)
SNAME		VARCHAR2(30)

COURSECODE	NUMBER(2)
STREET	VARCHAR2(30)
CITY	VARCHAR2(30)
PINCODE	NUMBER(6)
PHONE	NUMBER(10)
ENQUIRYDATE	NOT NULL DATE

SQL> desc Batch_ur14cs290;

Name	Null?	Type
BATCHCODE	NOT NULL	VARCHAR2(20)
COURSECODE	NOT NULL	NUMBER(2)
STARTINGDATE	NOT NULL	DATE
DURATION	NOT NULL	NUMBER(3)
COURSEFEES	NOT NULL	NUMBER(10,2)

SQL> desc Course_ur14cs290;

Name	Null?	Type
COURSECODE	NOT NULL	NUMBER(2)
COURSENAME		VARCHAR2(30)
SYLLABUS		VARCHAR2(60)

SQL> alter table Enquiry_ur14cs290 add en_st varchar2(20);

Table altered.

SQL> CREATE TABLE Enrollment_new(rollno number(8) constraint co9 not null, enquiryno number(8), batchcode number(5), enrollmentdate date constraint as2 not null);

Table created.

SQL> insert into enrollment_new select * from enrollment_ur14cs290;

0 rows created.

SQL> alter table Enrollment_ur14cs290 drop constraint unr1;

Table altered.

SQL> select * from Enquiry_ur14cs290 where enquirydate = sysdate;

no rows selected

SQL> create user floura identified by mypass;

User created.

SQL> grant insert,delete,update on Enrollment_ur14cs290 to floura;

Grant succeeded.

SQL> revoke all on Enrollment_ur14cs290 from floura;

Revoke succeeded.

SQL> savepoint s1;

Savepoint created.

SQL> alter table Enrollment_ur14cs290 rename column enrollmentdate to endate;

Table altered.

SQL> desc Enrollment_ur14cs290;

Name	Null?	Type
ROLLNO	NOT NULL	NUMBER(8)
ENQUIRYNO	NOT NULL	NUMBER(8)
BATCHCODE		NUMBER(5)
ENDATE	NOT NULL	DATE

SQL> delete from Enquiry_ur14cs290 where enquirydate = sysdate;

0 rows deleted.

SQL> delete from Enquiry_ur14cs290 where enquirydate = current_date;

0 rows deleted.

SQL> commit;

Commit complete.

SQL> rollback;

Rollback complete.

SQL> spool off;

EXPERIMENT 3:

Link: https://www.youtube.com/watch?v=L4g9UJWT63c

DESCRIPTION:

GROUP FUNCTIONS:

Group function work on a group of data to obtain aggregated values.

SELECT AVG(column_name) FROM table_name

SELECT COUNT(column_name) FROM table_name;

SELECT MIN(column_name) FROM table_name;

SELECT MAX(column_name) FROM table_name;

SELECT SUM(column_name) FROM table_name;

GROUP BY CLAUSE

This allows us to use simultaneous column name and group functions.

HAVING CLAUSE

This is used to specify conditions on rows retrieved by using group by clause.

CODE:

SQL> select avg(netincome) from Batch_ur14cs290 group by coursecode;

AVG(NETINCOME)

9000

8000

SQL> select coursename, avg(netincome) "maximum net income" from course_ur14cs290 group by coursecode having avg(netincome)>=all(select avg(netincome) from course_ur14cs290 group by coursecode);

COURSENAME MAXIMUM NET INCOME

---

java 9000

SQL> select amount "total feespaid" from feespaid_ur14cs290 where rollno=20002001;

total feespaid

5000

SQL> select c.coursename, b.coursecode,b.netincome from course_ur14cs290 c, batch_ur14cs290 b where c.coursecode = b.coursecode order by coursecode;

COURSENAME COURSECODE NETINCOME

---

oracle 10 8000

oracle 10 8000

oracle 10 8000

java 20 9000

SQL> select count(enquiryno) from enquiry_ur14cs290 where coursecode=10 order by coursecode;

COUNT(ENQUIRYNO)

1

SQL> select fname from enquiry_ur14cs290 where fname like 's%';

FNAME

---

sachin

SQL> select to_char(netincome, '99,999,99') "NET INCOME" from batch_ur14cs290;

(NET INCOME)

90,00

80,00

80,00

80,00

SQL> select initcap(fname) from enquiry_ur14cs290;

INITCAP(FNAME)

Anil

Sachin

SQL> select netincome*7 "WEEKLY INCOME" , netincome*30 "MONTHLY INCOME" from batch_ur14cs290;

WEEKLY INCOME	MONTHLY INCOME
63000	270000
56000	240000
56000	240000
56000	240000

SQL> select ('$'||amount) "AMOUNT" from feespaid_ur14cs290;

AMOUNT
$5000
$4000
$3000
$4000
$5000

SQL> select coursecode, max(count(coursecode)) "Maximum Enquiry" from enquiry_ur14cs290 group by coursecode;

COURSECODE	MAXIMUM ENQUIRY

```
        10              1

        20              1
```

```
SQL> select (fname||' '||sname) "NAME" from enquiry_ur14cs290 order by fname desc;

NAME
--------------------------------------------------------------------

sachin nimje

anil sharma
```

```
SQL> select avg(count(coursecode)) from course_ur14cs290 group by coursecode having
avg(coursecode)>all(select coursecode from course_ur14cs290 group by coursecode);

AVG(COUNT(COURSECODE))
----------------------
```

```
SQL> select min(coursefee) "MINIMUM PRICE" , coursecode from course_ur14cs290 group
by coursecode;

MINIMUM PRICE    COURSECODE
------------     ----------
        4000             10

SQL> commit;

Commit complete.

SQL> spool off;
```

EXPERIMENT 4:

Link: https://www.youtube.com/watch?v=QR5P2aSk1Yg&feature=youtu.be

DESCRIPTION:

Natural join matches tuples with the same values for all common attributes, and retains only one copy of each common column

Danger in natural join: beware of unrelated attributes with same name which get equated incorrectly

Set operations **union, intersect,** and **except**

Each of the above operations automatically eliminates duplicates

To retain all duplicates use the corresponding multiset versions **union all, intersect all**and **except all.**

Suppose a tuple occurs *m* times in *r* and *n* times in *s,* then, it occurs:

- *m* + *n* times in *r* **union all** *s*
- min(*m*,*n)* times in *r***intersect all** *s*
- max(0, *m* – *n)* times in *r***except all** *s*

CODE:

YOUTUBE LINK:

https://youtu.be/QR5P2aSk1Yg

SQL> select fname,sname from Enquiry_ur14cs290_second where city='CBE' union select sname,fname from Enquiry_ur14cs290_second where city='erd';

FNAME SNAME

---------- ----------

anil sharma

nimje sachin

SQL> select * from Enquiry_ur14cs290_second where enquirydate between '01-jan-14' and '01-dec-14';

ENQUIRYNO FNAME SNAME

---------- ---------- ----------

COURSECODE STREET CITY

---------- ---------- ----------

PINCODE PHONE ENQUIRYDA EMAILEN_ST

---------- ---------- --------- ----------

110001 anil sharma

10 ramnagar cbe

641114 9486453331 20-JAN-14

110002 sachin nimje

20 gandhi nagar erd

641142 9234567890 25-OCT-14

ENQUIRYNO FNAME SNAME

---------- ---------- ----------

COURSECODE STREET CITY

---------- ---------- ----------

PINCODE PHONE ENQUIRYDA EMAILEN_ST

---------- ---------- --------- ----------

SQL> select e.fname,c.coursename,e.city from Enquiry_ur14cs290_second e natural join course_ur14cs290 c where c.coursename='Oracle' and e.city='CBE';

FNAME COURSENAME CITY

---------- ---------- ----------

anil oracle cbe

SQL> select coursecode,fname from Enquiry_ur14cs290_second group by coursecode having count(coursecode)>1;

no rows selected

SQL> select e.fname,e.sname,e.city from course_ur14cs290 c,Enquiry_ur14cs290_second e where c.coursecode=e.coursecode and c.coursename<>'java';

FNAME SNAME

---------- ----------

anil sharma

SQL> select coursecode from course_ur14cs290 group by coursecode having count(coursecode)=0;

no rows selected

SQL> Select distinct fname,sname,city,enquiryno from Enquiry_ur14cs290_second;

FNAME SNAME

---------- ----------

CITY ENQUIRYNO

---------- ----------

anil sharma

cbe 110001

sachin nimje

erd 110002

SQL> Select fname,sname,city,enquiryno from Enquiry_ur14cs290_second;

FNAME SNAME

---------- ----------

CITY ENQUIRYNO

---------- ----------

anil sharma

cbe 110001

sachin nimje

erd 110002

SQL> select coursecode from Enquiry_ur14cs290_second group by coursecode having count(coursecode)> avg(coursecode);

no rows selected

SQL> select e.rollno,i.enquiryno,e.enrollmentdate,f.feespaiddate,f.chequeno,f.bankname from enrollment_ur14cs290 e,Enquiry_ur14cs290_second i, feespaid_ur14cs290 f where e.rollno<>0 and f.amount<>0 and e.rollno=f.rollno;

ROLLNO	ENQUIRYNO	ENROLLMEN	FEESPAIDD	CHEQUENO	BANKNAME
20002001	110001	01-JUL-13	01-JUL-13	123456	hdfc
20002002	110002	25-OCT-13	25-OCT-13		
20002003	110021	11-JAN-14	11-JAN-14	876544	tmb
20002004	110022	29-JAN-14	29-JAN-14	765433	dbi
20002005	110023	01-MAR-13	01-MAR-14		

EXPERIMENT 5:

Link: https://www.youtube.com/watch?v=LX85vEh9JGw&feature=youtu.be

DESCRIPTION:

CODE:

SQL> select fname from Enquiry_ur14cs290_second where enquiryno in (select enquiryno from enrollment_ur14cs290);

FNAME

Anil

Sachin

SQL> select * from Enquiry_ur14cs290_second where enquirydate between '01-jan-14' and '01-dec-14';

ENQUIRYNO FNAME SNAME

COURSECODE STREET CITY

PINCODE PHONE ENQUIRYDA E

110001 Anil Sharma

10 Ramnagar CBE

641114 9486453331 20-JAN-14 Y

110002 Sachin Nimje

20 Gandhi Nagar Gandhi Nagarerd

641142 9234567890 25-NOV-14 N

ENQUIRYNO FNAME SNAME

---------- ---------- ----------

COURSECODE STREET CITY

---------- ---------- ----------

PINCODE PHONE ENQUIRYDA E

---------- ---------- --------- -

SQL> select fname from Enquiry_ur14cs290_second where coursecode = (select coursecode from course_ur14cs290 where coursecode=10);

FNAME

Anil

SQL> insert into Enquiry_ur14cs290_second values (&eno,'&fname','&sname',&coursecode,'&street','&city',&pincode,&phone,'&enquiry','&en');

Enter value for eno: 110003

Enter value for fname: Floura

Enter value for sname: Angel

Enter value for coursecode: 10

Enter value for street: Malad

Enter value for city: Bombay

Enter value for pincode: 640097

Enter value for phone: 1001001003

Enter value for enquiry: 07-Nov-14

Enter value for en: Y

old 1: insert into Enquiry_ur14cs290_second values
(&eno,'&fname','&sname',&coursecode,'&street','&city',&pincode,&phone,'&enquiry','&en')

new 1: insert into Enquiry_ur14cs290_second values
(110003,'Floura','Angel',10,'Malad','Bombay',640097,1001001003,'07-Nov-14','Y')

1 row created.

SQL> /

Enter value for eno: 110004

Enter value for fname: Raja

Enter value for sname: haji

Enter value for coursecode: 20

Enter value for street: Borivali

Enter value for city: Bombay

Enter value for pincode: 640097

Enter value for phone: 2001003004

Enter value for enquiry: 14-Dec-14

Enter value for en: N

old 1: insert into Enquiry_ur14cs290_second values
(&eno,'&fname','&sname',&coursecode,'&street','&city',&pincode,&phone,'&enquiry','&en')

```
new   1: insert into Enquiry_ur14cs290_second values
(110004,'Raja','haji',20,'Borivali','Bombay',640097,2001003004,'14-Dec-14','N')


1 row created.


SQL> select fname from Enquiry_ur14cs290_second where coursecode = (select  coursecode
from  course_ur14cs290 where coursename='Java');


FNAME

------------------------------

Sachin

Raja


SQL> select * from Enquiry_ur14cs290_second;


 ENQUIRYNO FNAME                     SNAME

---------- ------------------------------ ------------------------------

COURSECODE STREET                      CITY

---------- ------------------------------ ------------------------------

  PINCODE      PHONE ENQUIRYDA E

---------- ---------- --------- -

    110001 Anil                        Sharma

       10 Ramnagar                     CBE

    641114 9486453331 20-JAN-14 Y
```

110002 Sachin Nimje

20 Gandhi Nagar Gandhi Nagarerd

641142 9234567890 25-NOV-14 N

ENQUIRYNO FNAME SNAME

---------- ---------- ----------

COURSECODE STREET CITY

---------- ---------- ----------

PINCODE PHONE ENQUIRYDA E

---------- ---------- --------- -

110003 Floura Angel

10 Malad Bombay

640097 1001001003 07-NOV-14 Y

110004 Raja haji

20 Borivali Bombay

ENQUIRYNO FNAME SNAME

---------- ---------- ----------

COURSECODE STREET CITY

---------- ---------- ----------

PINCODE PHONE ENQUIRYDA E

---------- ---------- --------- -

640097 2001003004 14-DEC-14 N

SQL> select fname from Enquiry_ur14cs290_second where city='Bombay' and coursecode= (select coursecode from course_ur14cs290 where coursename='Oracle');

FNAME

Floura

SQL> select fname from Enquiry_ur14cs290_second en1 where exists (select en2.city from Enquiry_ur14cs290_second en2 where en2.fname='Raja'and en1.city=en2.city);

FNAME

Raja

Floura

SQL> select fname from Enquiry_ur14cs290_second where exists (select coursecode from Enquiry_ur14cs290_second where coursecode =10 and coursecode =20);

no rows selected

SQL> select coursename from course_ur14cs290 where exists (select coursecode from Enquiry_ur14cs290_second where coursecode =10 and coursecode =20);

no rows selected

SQL> select fname from Enquiry_ur14cs290_second where city='CBE' and coursecode = (select coursecode from batch_ur14cs290 where batchcode=101);

no rows selected

SQL> select fname from Enquiry_ur14cs290_second where city='Bombay' and coursecode = (select coursecode from batch_ur14cs290 where batchcode=101);

FNAME

Raja

SQL> spool off;

EXPERIMENT 6:

Link : https://www.youtube.com/watch?v=cqXyqJt6VnA&feature=youtu.be

DESCRIPTION:

A **view** provides a mechanism to hide certain data from the view of certain users.

Any relation that is not of the conceptual model but is made visible to a user as a "virtual relation" is called a **view**.

A view is defined using the **create view** statement which has the form

create view v **as** <query expression >

where<query expression> is any legal SQL expression. The view name is represented by v.

CODE:

SQL> create view view1 as select coursecode as id_no,coursename as cname,netincome from course_ur14cs290;

View created.

SQL> select * from view1;

ID_NO	CNAME	NETINCOME
10	oracle	8000
20	java	9000

```
SQL> create or replace view view1 as select coursecode as id_no,coursename as
cname,netincome from course_ur14cs290 order by coursename;

View created.

SQL> select * from view1;

    ID_NO CNAME                          NETINCOME
---------- ------------------------------ ----------
        20 java                             9000
        10 oracle                           8000

SQL> create view view2 as select c.coursename,b.coursefees,b.startingdate,b.batchcode from
course_ur14cs290 c,batch_ur14cs290 b where c.coursecode=b.coursecode;

View created.

SQL> select * from view2;

COURSENAME                     COURSEFEES STARTINGD  BATCHCODE
------------------------------ ---------- --------- ----------
java                                 5000 01-JAN-16        101
```

oracle	4000	20-JAN-14	102
oracle	4000	01-APR-14	103
oracle	4000	25-OCT-14	104

SQL> create view view3 as select coursecode from course_ur14cs290 where coursename='Java' with check option;

View created.

SQL> select * from view3;

COURSECODE

20

SQL> spool off;

EXPERIMENT 7:

LINK: https://www.youtube.com/watch?v=DbiLtB2ztao

DESCRIPTION:

TRIGGER

- A trigger is a block of statements, which are fired automatically when a DML operation takes place.
- These are always associated with the database tables.

Triggers are classified as two types:

1. Row Level Triggers.
2. Statement Level Triggers.

Syntax of a Trigger

CREATE OR REPLACE TRIGGER <trigger name>

before/after<event1 or event2 or event3>

on<table>

[for each row[when <condition>]]

[DECLARE

<declarations>;]

BEGIN

<action block>;

END;

/

CODE:

```
SQL> create or replace trigger reregister
  2  before insert on enquiry_ur14cs290
  3  for each row
  4  declare
  5   code enquiry_ur14cs290.coursecode%type;
  6   begin
  7   select coursecode into code from enquiry_ur14cs290 where coursecode=10;
  8   if(code=10)
  9   then
 10    raise_application_error(-20110,'already registered');
 11     end if;
 12    end;
 13  /


SQL> insert into enquiry_ur14cs290
values(110004,'floura','angel',10,'karunya','cbe',641114,1001001001,'25-oct-16','y');

insert into enquiry_ur14cs328
values(110004,'floura','angel',10,'karunya','cbe',641114,1001001001,'25-oct-16','y');
            *
ORA-20110: already registered


SQL> create or replace trigger deletet
  2   before delete on enquiry_ur14cs290
  3   for each row
```

```
 4  declare
 5  begin
 6  if deleting then
 7      delete from enrollment_ur14cs290
 8      where enrollment_ur14cs290.enquiryno = :old.enquiryno;
 9  dbms_output.put_line('row is deleted with enquiryno: '||:old.enquiryno);
10   end if;
11   end;
12  /
```

Trigger created.

SQL> select * from Enrollment_ur14cs290;

```
   ROLLNO  ENQUIRYNO  BATCHCODE ENROLLMEN
---------- ---------- ---------- ---------
 20002001     110001        101 01-JUL-13
 20002002     110002        102 25-OCT-13
 20002003     110021        103 11-JAN-14
 20002004     110022        104 29-JAN-14
 20002005     110023        110 01-MAR-14
```

SQL> delete from enquiry_ur14cs290 where enquiryno = 110001;

row is deleted with enquiryno: 110001

```
SQL> select * from Enrollment_ur14cs290;

   ROLLNO  ENQUIRYNO  BATCHCODE ENROLLMEN
---------- ---------- ---------- ---------
  20002002     110002        102 25-OCT-13
  20002003     110021        103 11-JAN-14
  20002004     110022        104 29-JAN-14
  20002005     110023        110 01-MAR-14

SQL> create or replace trigger updatet
  2   before update on batch_ur14cs290
  3   for each row
  4   declare
  5   begin
  6   if updating then
  7    dbms_output.put_line('you can update the fees at the last day of the current year only:
'||:new.coursefees);
  8    end if;
  9    end;
 10  /

Trigger created.

SQL> select * from batch_ur14cs290;
```

BATCHCODE COURSECODE STARTINGD DURATION COURSEFEES NETINCOME

---------- ---------- --------- ---------- ---------- ----------

101 20 01-JAN-14 30 5000

102 10 20-JAN-14 20 4000

103 10 01-APR-14 20 4000

104 10 25-OCT-14 20 4000

SQL> update batch_ur14cs290 set coursefees=6000 where coursecode=20;

you can update the fees at the last day of the current year only: 6000

1 row updated.

SQL> select * from batch_ur14cs290;

BATCHCODE COURSECODE STARTINGD DURATION COURSEFEES NETINCOME

---------- ---------- --------- ---------- ---------- ----------

101 20 01-JAN-14 30 6000

102 10 20-JAN-14 20 4000

103 10 01-APR-14 20 4000

104 10 25-OCT-14 20 4000

SQL> declare

2 count number(9);

```
  3  cursor cur is select avg(coursefees) from batch_ur14cs290 group by coursecode;
  4   begin
  5   open cur;
  6  fetch cur into count;
  7   dbms_output.put_line('average is: '||count);
  8   close cur;
  9   end;
 10  /
  dbms_output.put_line('average is: '||count);
                              *
ERROR at line 7:
ORA-06550: line 7, column 41:
PLS-00204: function or pseudo-column 'COUNT' may be used inside a SQL statement
only
ORA-06550: line 7, column 4:
PL/SQL: Statement ignored


SQL> declare
  2  counttot number(9);
  3  cursor cur is select avg(coursefees) from batch_ur14cs290 group by coursecode;
  4   begin
  5   open cur;
  6  fetch cur into counttot;
  7   dbms_output.put_line('average is: '||counttot);
```

```
 8     close cur;
 9     end;
10  /
```

average is: 4500

PL/SQL procedure successfully completed.

```
SQL> DECLARE
 2     bcode enrollment_ur14cs290.batchcode%type;
 3    CURSOR c2 is
 4       SELECT batchcode into bcode from enrollment_ur14cs290 where rollno=20002004;
 5    BEGIN
 6    OPEN c2;
 7
 8     LOOP
 9    update  batch_ur14cs290 set coursefees=coursefees-(0.2*coursefees) where
batchcode=bcode;
10         FETCH c2 into bcode;
11         EXIT WHEN c2%notfound;
12         dbms_output.put_line('table is modified '||bcode);
13      END LOOP;
14      CLOSE c2;
15    END;
16  /
```

table is modified 101

table is modified 102

table is modified 103

PL/SQL procedure successfully completed.

SQL> select * from batch_ur14cs290;

BATCHCODE	COURSECODE	STARTINGD	DURATION	COURSEFEES	NETINCOME
101	20	01-JAN-14	30	4800	
102	10	20-JAN-14	20	4000	
103	10	01-APR-14	20	4000	
104	10	25-OCT-14	20	3200	

SQL> spool off;

EXPERIMENT 8:

LINK: https://www.youtube.com/watch?v=cnw7-kJroeI

DESCRIPTION:

Syntax of a function (ORACLE).

```
CREATE OR REPLACE FUNCTION <function name>[(
        <parameter1> <data type> [<width>],
        <parameter2> <data type> [<width>],
        <parameter3> <data type> [<width>],
        <parameter4> <data type> [<width>],
        .
        .
        .
        <parameterN> <data type> [<width>]
        )]

)
return <type>
as


BEGIN
      <function body>;.
      return <value>;
END;
/
```

Syntax of a Stored Procedure (ORACLE)

```
CREATE [OR REPLACE] PROCEDURE <procedure name>[(
        <parameter1> <data type> [<width>] [<parameter mode>],
        <parameter2> <data type> [<width>] [<parameter mode>],
        <parameter3> <data type> [<width>] [<parameter mode>],
        <parameter4> <data type> [<width>] [<parameter mode>],
        .
        .
        .
        <parameterN> <data type> [<width>] [<parameter mode>]
        )]
AS
        <DECLERATIONS>;
BEGIN
        <action-code>;
[EXCEPTION
        <action-code>;]
END;
```

CODE:

```
SQL> create or replace procedure f1(cid in number)
  2  is
  3  name enquiry_ur14cs290.fname%type;
  4  begin
  5  select fname into name from enquiry_ur14cs290 where coursecode=cid;
  6  dbms_output.put_line('name is '||name);
  7  end;
  8 /

Procedure created.

SQL> declare
  2  begin
  3  f1(11001);
```

```
  4    end;
  5  /
name is anil

PL/SQL procedure successfully completed.



SQL> create or replace procedure f2(dt in date, eno in number)
  2   is
  3  num batch_ur14cs290.duration%type;
  4    edate batch_ur14cs290.startingdate%type;
  5     begin
  6   select duration into num from batch_ur14cs290 where coursecode=eno;
  7    select startingdate into edate from batch_ur14cs290 where coursecode=eno;
  8  dbms_output.put_line('duration  is ' || num);
  9  dbms_output.put_line('starting date   is ' || edate);
 10   end;
 11  /

Procedure created.

SQL> declare
  2   begin
  3   f2('01-jul-13',20);
  4   end;
```

```
  5  /

duration  is 30
starting date   is 01-JAN-14

PL/SQL procedure successfully completed.

SQL> create or replace procedure f3(rol in number)
  2     is
  3     begin
  4        delete from enrollment_ur14cs290 where rollno=rol;
  5    exception
  6    when
  7   NO_DATA_FOUND then raise_application_error(-20100,'the given data is not present');
  8     end;
  9  /

Procedure created.

SQL> create or replace function f4(fee in number)
  2     return number is
  3   fees number(8);
  4    begin
  5     select b.coursefees into fees from batch_ur14cs290 b ,course_ur14cs290 c where
b.coursecode=fee and rownum=1;
```

```
  6     return fees;
  7    end;
  8  /
```

Function created.

```
SQL> declare
  2   numb number(8);
  3   begin
  4   numb:=f4(10);
  5   dbms_output.put_line('fees of the coursecode 10 is ' || numb);
  6    end;
  7  /
```

PL/SQL procedure successfully completed.

```
SQL> declare
  2  numb number(8);
  3  begin
  4  numb:=f4(10);
  5  dbms_output.put_line('fees of the coursecode 10 is ' || numb);
  6  end;
  7  /
```

fees of the coursecode 10 is 4000

```
SQL> create or replace function f5
  2   return number is
  3   num number(2);
  4  begin
  5   select count(c.coursecode) into num from course_ur14cs290 c ,feespaid_ur14cs290 f having
f.amount=5000;
  6   return num;
  7  end;
  8  /

Function created.

SQL> declare
  2  numb number(2);
  3  begin
  4  numb:=f5();
  5  dbms_output.put_line('number of course whose amount is 5000 is ' || numb);
  6  end;
  7  /
number of course whose amount is 5000 is 1

SQL> create or replace function f6(roll in number)
  2    return varchar2 is
  3    nf enquiry_ur14cs290.fname%type;
  4   begin
```

```
  5      select e.fname into nf from enquiry_ur14cs290 e, enrollment_ur14cs290 e1 where
e1.enquiryno=roll and rownum=1;

  6     return nf;

  7     end;

  8  /
```

Function created.

```
SQL> declare

  2  nm varchar2(20);

  3  begin

  4  nm:=f6(110001);

  5  dbms_output.put_line('fname is '||nm);

  6  end;

  7  /
```

fname is anil

PL/SQL procedure successfully completed.

```
SQL> spool off;
```

EXPERIMENT-9: JDBC

LINK: https://www.youtube.com/watch?v=iSZBv_hAFTM

AIM:

To set up a connection between java and databasae

DESCRIPTION:

PROCEDURE:

STEP 1: create a project in netbeans ide 8.0.2

STEP 2: go to libraries and add the jar file/folder

STEP 3: go to services->database->right click on jdbc connectivity-> click connect and connect the project with the database

STEP 4: type the required code and give the appropriate username and password

CODE:

```
package jdbc;


import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Scanner;


/**

 *

 * @author Floura

 */
```

```
public class Main {


   /**
    * @param args the command line arguments
    */
   public static void main(String[] args) {
      // TODO code application logic here
      {


      ResultSet r;
      try
{
Class.forName("oracle.jdbc.driver.OracleDriver")   ;
Connection c=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","floura");
Statement st=c.createStatement();
r=st.executeQuery("select * from course_ur14cs290");
Scanner s=new Scanner (System.in);
while(true){
System.out.println("Enter your choice");
  System.out.println("1.INSERT");
  System.out.println("2.UPDATE");
  System.out.println("3.DELETE");
 int ch=s.nextInt();
 switch(ch){
    case 1: System.out.println("Enter codeno");
```

```
int code= s.nextInt();
  System.out.println("Enter coursename");
String name= s.next();
  System.out.println("Enter syllabus");
String syl= s.next();
//  System.out.println("Enter amount");
//int amt= s.nextInt();
String query="insert into course_ur14cs290 "+"values (" +code+",'"+name+"','"+syl+"')";
st.executeUpdate(query);
        System.out.println("Value Inserted");
break;


     case 2:
        System.out.println("Enter the coursename you need to update");
        String cn=s.next();
         System.out.println("Enter the courscode where you need to update");
         int cc=s.nextInt();
         String query1="update course_ur14cs290"+" set coursename ="+"'"+cn+"'"+"where coursecode
="+cc;
         st.executeUpdate(query1);
         System.out.println("Table Updated");
         break;


     case 3:
        System.out.println("Enter the coursecode you want to delete");
        int dc=s.nextInt();
```

```
        String query2="delete from course_ur14cs290 where coursecode ="+dc;
        st.executeUpdate(query2);
        System.out.println("Record Deleted");
        break;

     default:
        System.out.println("Wrong Choice");
  }
while(r.next())
System.out.println(r.getInt(1)+"\t"+r.getString(2));

}
}
catch(SQLException e)
{
System.out.println("SQL error:"+e);
}
catch(Exception e)
{
System.out.println("Error:"+e);
}
}
   }

}
```

OUPUT:

RESULT: The connection has been successfully set up

EXPERIMENT: 10

LINK: https://youtu.be/tS3bEJI7Qlg

Aim : Make an application in Android Application(Karunya Map)

Code: Database Handler

```
package com.example.allie.map;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.SQLException;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.widget.Toast;

import static com.example.allie.map.R.id.sno;


/**

* Created by floura  and Priya on 29/10/16.

*/

public class DatabaseHandler extends SQLiteOpenHelper {


//columns

public static final String DATABASE_NAME ="Details";

public static final String AUTO_ID = "AUTO_ID";

public static final String TABLE_NAME = "KarunyaMap";

public static final String ID = "Surveyno";

public static final String NAME = "Block";
```

```
public static final String PERSON_INCHARGE = "Person_name";

public static final String INFO = "complete_info";


Context mcontext;


public DatabaseHandler(Context context) {

super(context, DATABASE_NAME, null, 1);

this.mcontext = context;

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL("create table " + TABLE_NAME

+"(" + ID + " text , " + AUTO_ID + " integer primary key autoincrement, " + NAME  + " text, "
+ PERSON_INCHARGE+ " text, "

+ INFO  + " text); "  );

}


@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL("DROP TABLE IF EXIST"+TABLE_NAME);

onCreate(db);

}


public long INSERT_DATA(String sno, String block, String pi, String details)

{
```

```
SQLiteDatabase db = this.getWritableDatabase();
ContentValues contentValues = new ContentValues();
contentValues.put(NAME,block);
contentValues.put(ID, sno);
contentValues.put(PERSON_INCHARGE,pi);
contentValues.put(INFO, details);
return  db.insert(TABLE_NAME,null,contentValues);
}

public Boolean UPDATE_DATA(String sno, String block, String pi, String details)
{
SQLiteDatabase db = this.getWritableDatabase();
ContentValues contentValues = new ContentValues();
contentValues.put(NAME,block);
contentValues.put(ID, sno);
contentValues.put(PERSON_INCHARGE,pi);
contentValues.put(INFO, details);
return db.update(TABLE_NAME,contentValues,ID+"="+sno,null) > 0;
}

public boolean DELETE_DATA(String sno)
{
SQLiteDatabase db = this.getWritableDatabase();
return db.delete(TABLE_NAME, ID + "=" + sno, null)>0;
}
```

```
public Cursor GET_ALL_DATA()
{
SQLiteDatabase db = this.getWritableDatabase();
return db.rawQuery("select" +ID+"  As _id ,* from "+TABLE_NAME,null);
}
public Cursor GET(String row_id)throws SQLException
{
SQLiteDatabase db = this.getWritableDatabase();
Cursor cursor = db.query(true, TABLE_NAME, new String[]{ID,
NAME,PERSON_INCHARGE,INFO
}, row_id + "=" + ID, null, null, null, null, null);
if(cursor!=null)
{
cursor.moveToNext();
}
return cursor;
}

public boolean check_value(String sno){
SQLiteDatabase db = this.getWritableDatabase();
String Query = "Select * from " + TABLE_NAME + " where " + ID + " = '" + sno+"'";
Cursor cursor = db.rawQuery(Query, null);
if(cursor.getCount() <= 0){
cursor.close();
```

```
return false;
}
return true;
}
}
```

Main Activity 2

```
package com.example.allie.map;
import android.app.AlertDialog;
import android.content.Intent;
import android.database.Cursor;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class Main2Activity extends AppCompatActivity {
    EditText editText;
    Button button, button2;
    DatabaseHandler handler;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
```

```
    handler = new DatabaseHandler(this);
    editText = (EditText)findViewById(R.id.snum);
    button = (Button)findViewById(R.id.abt);
    button2 = (Button)findViewById(R.id.vbt);
    registerbutton();

}

private void registerbutton()
{
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String sno = editText.getText().toString();
            if (sno.length()==0)
            {
                editText.setError("Required!!!");
                Toast.makeText(getApplicationContext(),"Survey Number
Required",Toast.LENGTH_SHORT).show();
            }
            else {
                if (handler.check_value(sno)) {
                    Intent intent = new Intent(getApplicationContext(), View2.class);
                    intent.putExtra(handler.ID, sno);
                    startActivity(intent);
```

```
                Toast.makeText(getApplicationContext(),"Details already exists. Edit
details",Toast.LENGTH_SHORT).show();


            } else {
                Intent intent = new Intent(getApplicationContext(), Add.class);
                intent.putExtra(handler.ID, sno);
                startActivity(intent);
            }
        }
    }
});


    button2.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String sno = editText.getText().toString();
            if (sno.length()==0)
            {
                editText.setError("Required!!!");
                Toast.makeText(getApplicationContext(),"Survey Number
Required",Toast.LENGTH_SHORT).show();
            }
            else {
                if(handler.check_value(sno)) {
                    Intent intent = new Intent(getApplicationContext(), View2.class);
```

```
                intent.putExtra(handler.ID, sno);
                startActivity(intent);
            }
            else{
                Intent intent = new Intent(getApplicationContext(), Add.class);
                intent.putExtra(handler.ID, sno);
                startActivity(intent);
                Toast.makeText(getApplicationContext(),"No Details, Add new
Details",Toast.LENGTH_SHORT).show();
            }
        }
    }
});
}
public void show(String title,String message)
{
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setCancelable(true);
    builder.setTitle(title);
    builder.setMessage(message);
    builder.show();
}
}
```

NewActivity

```
package com.example.allie.map;
```

```
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.TimePickerDialog;
import android.content.DialogInterface;
import android.database.Cursor;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.TimePicker;
import android.widget.Toast;

import java.text.SimpleDateFormat;
import java.util.Calendar;

public class NewActivity extends AppCompatActivity {

EditText ID,PI,INFO;
```

Button Update,Delete,Add,ViewD, see;

Spinner Blocks;

DatabaseHandler handler;

private long ROW_ID;

final String cont[] = {

"Aerodynamics Lab-I",

"Angel Garden",

"Angelina Block",

"Animal House",

"Bank Building CUM Economical GH",

"Bethany(BR) Hostel",

"Bethesda",

"BRR Hostel",

"Carmel Staff Quarters",

"CCD",

"Civil Engineering Dept",

"Computer Centre",

"CST Block",

"Dhakshanamoorthy Block",

"DoVE Building",

"Ebenezer Auditorium",

"ECE",

"Edward George Block",

"Elohim Auditorium",

"Emmanuel Auditorium",

"Evangeline/OPHRA",

"Father Duraisamy Hostel",

"Generator cum Power Room",

"Hepzibah Block",

"JMR Hostel",

"JVR Hostel",

"Kidron Staff Quarters",

"KIS School Building",

"Kitchen & Dining Block",

"MBA Block",

"Mechanical Engineering",

"New Electrical Sciences",

"New Science & Humanities",

"Old Electrical Sciences",

"Old Science & Humanities",

"P R Garg Residence",

"Post Office",

"Senior Bachelor's Quarters",

"Servant Quarters (Guest House)",

"Sevugapandian Block",

"Sewage Treatment Plant",

"Sewage Treatment Plant",

"Seesha",

"Stores",

"Students Amenity Centre",

```
"Sundararaj Hostel",
"Vistors Lounge in South Block",
};
int contr;


@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_new);
PI = (EditText)findViewById(R.id.pi);
INFO= (EditText)findViewById(R.id.details);
ID = (EditText)findViewById(R.id.sno);
Update = (Button)findViewById(R.id.update);
Delete = (Button)findViewById(R.id.delete);
Add = (Button)findViewById(R.id.add);
ViewD = (Button)findViewById(R.id.viewdetails);
Blocks =(Spinner)findViewById(R.id.spinner2);
see = (Button)findViewById(R.id.see);
handler = new DatabaseHandler(this);
if (savedInstanceState != null) ROW_ID =
savedInstanceState.getLong(DatabaseHandler.AUTO_ID);
else ROW_ID = 0;
SET_SPINNER();
REGISTER_BUTTON_LISTENER();
}
```

```
private void SET_SPINNER()
{
ArrayAdapter<String> arrayAdapter = (new
ArrayAdapter<String>(getApplicationContext(),R.layout.spinnerview,cont){
@Override
public View getView(int position, View convertView, ViewGroup parent) {
return super.getView(position, convertView, parent);
}
});
Blocks.setAdapter(arrayAdapter);
Blocks.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
@Override
public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
contr = position;
}

@Override
public void onNothingSelected(AdapterView<?> parent) {

}
});
}

protected void onResume() {
```

```
super.onResume();
SET_ROW_ID_FROM_INTENT();
}

private void SET_ROW_ID_FROM_INTENT()
{
if(ROW_ID == 0)
{
Bundle bundle = getIntent().getExtras();
if (bundle != null) ROW_ID = bundle.getLong(handler.ID);
else ROW_ID = 0;
}
}

private void REGISTER_BUTTON_LISTENER()
{
ViewD.setOnClickListener(new View.OnClickListener(){
@Override
public void onClick(View v) {
Toast.makeText(getApplicationContext(), "Button Clicked", Toast.LENGTH_SHORT).show();
GET_INFO();
}
});
Update.setOnClickListener(new View.OnClickListener() {
@Override
```

```
public void onClick(View v) {
Toast.makeText(getApplicationContext(), "Button Clicked", Toast.LENGTH_SHORT).show();
UPDATE_DETAILS();
}
});

Delete.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
DELETE_STATE();
}
});

Add.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
SAVE_STATE();
}
});

//to confirm if data is being added or not
see.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Cursor cursor = handler.GET_ALL_DATA();
```

```
StringBuffer buffer = new StringBuffer();
while(cursor.moveToNext())
{
buffer.append(cursor.getString(0)+"\n");
buffer.append(cursor.getString(3)+"\n\n");
}
show("Data",buffer.toString());
}
});
}

public void show(String title,String message)
{
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setCancelable(true);
builder.setTitle(title);
builder.setMessage(message);
builder.show();
}

private void GET_INFO(){
int flag = 1;
final String survey = ID.getText().toString();
if (survey.length()==0)
{
```

```
ID.setError("Required!!!");

Toast.makeText(getApplicationContext(),"Survey Number
Required",Toast.LENGTH_SHORT).show();

flag=0;

}

if(flag!=0) {

if (ROW_ID == 0) {

handler.GET(survey);

}

}

}


private void UPDATE_DETAILS(){

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setMessage("Are you sure you want to update?")

.setCancelable(false)

.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int id) {

String pname = PI.getText().toString();

String sno= ID.getText().toString();

String details = INFO.getText().toString();

String blocks = cont[contr];

handler.UPDATE_DATA(sno,blocks,pname,details);

}

})
```

```
.setNegativeButton("No", new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int id) {
dialog.cancel();
}
});
AlertDialog alert = builder.create();
alert.show();
finish();
Toast.makeText(getApplicationContext(), "Details Updated", Toast.LENGTH_SHORT).show();
}

private void DELETE_STATE(){
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setMessage("Are you sure you want to delete?")
.setCancelable(false)
.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int id) {
String sno= ID.getText().toString();
handler.DELETE_DATA(sno);
}
})
.setNegativeButton("No", new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int id) {
dialog.cancel();
}
```

```
});
AlertDialog alert = builder.create();
alert.show();
finish();
Toast.makeText(getApplicationContext(), "Details Deleted", Toast.LENGTH_SHORT).show();
}



private void SAVE_STATE()
{
int flag = 1;
String pname = PI.getText().toString();
if(pname.length()==0)
{
PI.setError("Required!!!");
Toast.makeText(getApplicationContext(),"Person Incharge
Required",Toast.LENGTH_SHORT).show();
flag=0;
}


String sno= ID.getText().toString();
if (sno.length()==0)
{
ID.setError("Required!!!");
```

```
Toast.makeText(getApplicationContext(),"Survey Number
Required",Toast.LENGTH_SHORT).show();

flag=0;

}


String details = INFO.getText().toString();

if(details.length()==0)

{

INFO.setError("Required!!!");

Toast.makeText(getApplicationContext(),"Details Required",Toast.LENGTH_SHORT).show();

flag=0;

}

String blocks = cont[contr];

if(flag!=0)

{

if (ROW_ID == 0) {

long id = handler.INSERT_DATA(sno,blocks,pname,details);

if (id > 0) {

ROW_ID = id;

}

}

finish();

Toast.makeText(getApplicationContext(), "Details Added", Toast.LENGTH_SHORT).show();

}

}
```

```
@Override
protected void onStop() {
// if(reminderManager!=null)
//   reminderManager.doUnbindService();
super.onStop();
}

@Override
public void onSaveInstanceState(Bundle outState) {
super.onSaveInstanceState(outState);
outState.putLong(DatabaseHandler.ID,ROW_ID);
}
}
```

MAPSACTIVITY

```
package com.example.allie.map;
import android.app.AlertDialog;
import android.content.Context;
import android.*;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Color;
import android.location.*;
import android.location.LocationListener;
import android.location.LocationManager;
```

```
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.*;
import android.widget.Spinner;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.MapView;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.UiSettings;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.PolylineOptions;
import android.widget.AdapterView.OnItemSelectedListener;

import java.io.IOException;
import java.util.List;

public class MapsActivity extends FragmentActivity implements OnItemSelectedListener{
DatabaseHandler handler;

private GoogleMap mMap; // Might be null if Google Play services APK is not available.
//MapView map = (MapView) findViewById(R.id.map);
```

```
String pla[]={
"Aerodynamics Lab-I",
"Angel Garden",
"Angelina Block",
"Animal House",
"Bank Building CUM Economical GH",
"Bethany(BR) Hostel",
"Bethesda",
"BRR Hostel",
"Carmel Staff Quarters",
"CCD",
"Civil Engineering Dept",
"Computer Centre",
"CST Block",
"Dhakshanamoorthy Block",
"DoVE Building",
"Ebenezer Auditorium",
"ECE",
"Edward George Block",
"Elohim Auditorium",
"Emmanuel Auditorium",
"Evangeline/OPHRA",
"Father Duraisamy Hostel",
"Generator cum Power Room",
"Hepzibah Block",
```

"JMR Hostel",

"JVR Hostel",

"Kidron Staff Quarters",

"KIS School Building",

"Kitchen & Dining Block",

"MBA Block",

"Mechanical Engineering",

"New Electrical Sciences",

"New Science & Humanities",

"Old Electrical Sciences",

"Old Science & Humanities",

"P R Garg Residence",

"Post Office",

"Senior Bachelor's Quarters",

"Servant Quarters (Guest House)",

"Sevugapandian Block",

"Sewage Treatment Plant",

"Sewage Treatment Plant",

"Seesha",

"Stores",

"Students Amenity Centre",

"Sundararaj Hostel",

"Vistors Lounge in South Block",

};

double lat[]={10.93254,

10.94294,

10.93965,

10.93571,

10.93955,

10.94314,

10.93579,

10.94004,

10.93604,

10.93714,

10.93537,

10.93520,

10.93361,

10.93785,

10.93531,

10.93439,

10.93635,

10.93960,

10.93475,

10.93397,

10.93505,

10.94029,

10.93670,

10.93916,

10.94066,

10.94123,

```
10.93553,
10.93902,
10.94219,
10.93454,
10.93471,
10.93621,
10.93632,
10.93604,
10.93439,
10.93812,
10.93976,
10.93804,
10.94019,
10.93738,
10.93823,
10.94394,
10.93859,
10.93441,
10.93729,
10.93647,
10.93699,
};
double lon[]={76.74508,
76.75306,
76.74328,
```

76.74277,

76.74589,

76.74558,

76.74473,

76.74505,

76.74003,

76.74403,

76.74405,

76.74326,

76.74305,

76.74006,

76.74267,

76.74370,

76.74296,

76.74114,

76.74313,

76.74493,

76.73867,

76.74129,

76.74076,

76.74333,

76.74505,

76.74511,

76.73993,

76.75448,

```
76.74532,
76.74305,
76.74403,
76.74176,
76.74114,
76.74410,
76.74484,
76.73942,
76.74587,
76.74106,
76.74610,
76.73974,
76.73943,
76.74649,
76.74174,
76.74532,
76.74015,
76.73957,
76.74042,
};
String dis[]={"PT/2009/MRP/2009-10 dt 04.12.2009",
"",
"PT/2001 dt 27.01.2001",
"PT/2010/MRP/2010-11 dt 06.01.2011",
"PT/2001 dt 12.01.2001",
```

"PT/2007/MRP/2006-07 dt 29.01.2007",

"-",

"PT/2007/MRP/2006-07 dt 29.01.2007",

"PT/1998 /dt 08.01.1998",

"-",

"PT/3475/87 A5/ dt 29.04.1987",

"PT/2002 dt 23.04.2002",

"PT/2002/ dt 24.02.2002",

"PT/2002/ dt 05.01.2002",

"PT/2007MRP/2006-07 dt 29.01.2007",

"PT/1987/ dt 13.08.1997",

"-",

"PT/1997 dt 15.06.1997",

"PT/2002 dt 23.04.2002",

"PT/2000 dt 21.07.2000",

"PT/2007/MRP/2006-2007 dt 29.01.2007",

"PT/1997 /dt 25.06.1997",

"PT/1997 dt 12.09.1997",

"PT/2000 dt 16.01.2000",

"PT/2007/MRP/2006-07 dt 29.01.2007",

"PT/2007/MRP/2006-07 dt 29.01.2007",

"PT/1997 dt 15.10.1997",

"PT/2002 /dt 23.04.2002",

"PT/2007/MRP/2006-07 dt 29.01.2007",

"PT/2002 dt 23.04.2002",

```
"PT/1987/ dt 13.08.1997",
"PT/2007/MRP/2006-07 dt 29.01.2007",
"PT/2007/MRP/2006-07 dt 29.01.2007",
"PT/5063/ 96 A3 dt 07.01.1997",
"PT/1997 dt 13.08.1997",
"PT/2007/MRP/2006-07 dt 29.01.2007",
"PT/2001/dt 12.01.2001",
"PT/2001/10.05.01 dt 10.05.2001",
"PT/2001 dt 21.03.2002",
"PT/2001/dt 16.07.2001",
"PT/2007/MRP/2006-2007 dt 29.01.2007",
"PT/2007/MRP/2006-07 dt 29.01.2007",
"-",
"PT/2011/MRP/2011-12 dt 19.08.2011",
"PT/2012/MRP/2012-13 dt 30.10.2012",
"PT/2001/dt 12.04.2001",
"PT/2001 dt 10.11.2001",
};
LatLng latLng;
String loc,item;
private UiSettings mUiSettings;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_maps);
//      Button addbtns = (Button) findViewById(R.id.addbtns);
//      addbtns.setOnClickListener(new View.OnClickListener(){
//
//          @Override
//          public void onClick(View v) {
//              Intent i = new Intent(MapsActivity.this,ku_db.class);
//              startActivity(i);
//          }
//      });



setUpMapIfNeeded();
Spinner spinner = (Spinner) findViewById(R.id.spinner);
spinner.setOnItemSelectedListener(this);
ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, pla);
int str1=pla.length;
int str2=dis.length;
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(
new LatLng(10.93604, 76.74410), 16));
mMap.addPolyline(new PolylineOptions().geodesic(true)
.add(new LatLng(10.93794, 76.74040))
.add(new LatLng(10.93819, 76.73894))
.add(new LatLng(10.93614, 76.73815))
```

```
.add(new LatLng(10.93447, 76.73825))
.add(new LatLng(10.93437, 76.73927))
.add(new LatLng(10.93526, 76.73951))
.add(new LatLng(10.93534, 76.74001))
.add(new LatLng(10.93604, 76.74021))
.add(new LatLng(10.93794, 76.74040)).width(3).color(Color.CYAN));

mMap.addPolyline(new PolylineOptions().geodesic(true)
.add(new LatLng(10.93814, 76.74136))
.add(new LatLng(10.93831, 76.74184))
.add(new LatLng(10.93715, 76.74168))
.add(new LatLng(10.93698, 76.74251))
.add(new LatLng(10.93768, 76.74265))
.add(new LatLng(10.93758, 76.74398))
.add(new LatLng(10.93856, 76.74418))
.add(new LatLng(10.93899, 76.74572))
.add(new LatLng(10.93812, 76.74571))
.add(new LatLng(10.93526, 76.74549))
.add(new LatLng(10.93524, 76.74581))
.add(new LatLng(10.93151, 76.74536))
.add(new LatLng(10.93117, 76.74325))
.add(new LatLng(10.93116, 76.74186))
.add(new LatLng(10.93395, 76.74204))
.add(new LatLng(10.93416, 76.74137))
.add(new LatLng(10.93581, 76.74166))
```

```
.add(new LatLng(10.93623, 76.74047))
.add(new LatLng(10.93814, 76.74136)).width(3).color(Color.YELLOW));

mMap.addPolyline(new PolylineOptions().geodesic(true)
.add(new LatLng(10.93857, 76.73874))
.add(new LatLng(10.94013, 76.73916))
.add(new LatLng(10.94013, 76.74017))
.add(new LatLng(10.94105, 76.74041))
.add(new LatLng(10.94114, 76.74205))
.add(new LatLng(10.94065, 76.74423))
.add(new LatLng(10.93869, 76.74405))
.add(new LatLng(10.93849, 76.74230))
.add(new LatLng(10.93827, 76.74104))
.add(new LatLng(10.93857, 76.73874)).width(3).color(Color.RED));

mMap.addPolyline(new PolylineOptions().geodesic(true)
.add(new LatLng(10.93857, 76.73874))
.add(new LatLng(10.94013, 76.73916))
.add(new LatLng(10.94013, 76.74017))
.add(new LatLng(10.94105, 76.74041))
.add(new LatLng(10.94114, 76.74205))
.add(new LatLng(10.94065, 76.74423))
.add(new LatLng(10.93869, 76.74405))
.add(new LatLng(10.93849, 76.74230))
.add(new LatLng(10.93827, 76.74104))
```

```
.add(new LatLng(10.93857, 76.73874)).width(3).color(Color.RED));

mMap.addPolyline(new PolylineOptions().geodesic(true)
.add(new LatLng(10.93953, 76.74421))
.add(new LatLng(10.93917, 76.74572))
.add(new LatLng(10.94011, 76.74902))
.add(new LatLng(10.94414, 76.74812))
.add(new LatLng(10.94349, 76.74499))
.add(new LatLng(10.94196, 76.74440))
.add(new LatLng(10.93953, 76.74421)).width(3).color(Color.MAGENTA));
spinner.setAdapter(dataAdapter);

mMap.addPolyline(new PolylineOptions().geodesic(true)
.add(new LatLng(10.94150, 76.75225))
.add(new LatLng(10.94495, 76.75204))
.add(new LatLng(10.94466, 76.75400))
.add(new LatLng(10.94221, 76.75378))
.add(new LatLng(10.94150, 76.75225)).width(3).color(Color.GREEN));
spinner.setAdapter(dataAdapter);

mMap.addPolyline(new PolylineOptions().geodesic(true)
.add(new LatLng(10.94083, 76.75097))
.add(new LatLng(10.94188, 76.75335))
.add(new LatLng(10.94137, 76.75346))
.add(new LatLng(10.94140, 76.75390))
```

```
.add(new LatLng(10.94208, 76.75398))
.add(new LatLng(10.94268, 76.75559))
.add(new LatLng(10.93830, 76.75615))
.add(new LatLng(10.93840, 76.75385))
.add(new LatLng(10.93931, 76.75114))
.add(new LatLng(10.94083, 76.75097)).width(3).color(Color.DKGRAY));
spinner.setAdapter(dataAdapter);

}

@Override
protected void onResume() {
super.onResume();
setUpMapIfNeeded();
// Toast.makeText(this,"Hello welcome back",Toast.LENGTH_SHORT);
}

@Override
public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
item = parent.getItemAtPosition(position).toString();
Toast.makeText(parent.getContext(), "Selected: " + item, Toast.LENGTH_LONG).show();

}

public void onadd(View v){
```

```
Intent i = new Intent(MapsActivity.this,Main2Activity.class);
startActivity(i);
}

public void onsrch(View view)
{
String loc=item;
int ad=0;
int tat=0;
Boolean flag=false;
for(int i=0;i<pla.length;i++)
{
if(flag==false)
{
if (loc.equals(pla[i]))
{
//ad=i;
tat=i;
flag=true;
break;
} else
{
flag=false;
}
}
```

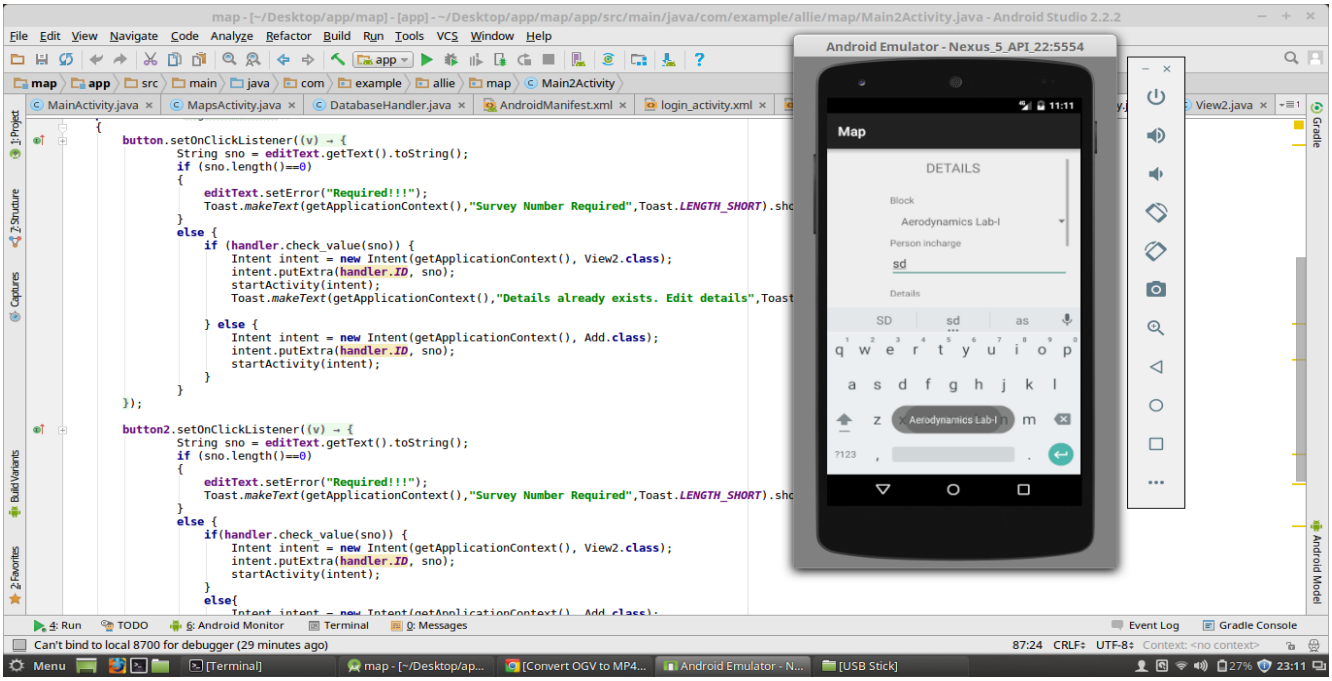
```
}
if(flag==true)
{
LatLng latLng = new LatLng(lat[tat], lon[tat]);
String detail;
detail = dis[tat];
//          detail = dis[tat];a
//mMap.addMarker(new MarkerOptions().position(latLng).title(loc).snippet("Population:
4,627,300").icon(BitmapDescriptorFactory.fromResource(R.drawable.cst)));
mMap.addMarker(new MarkerOptions().position(latLng).title(loc).snippet(detail));
mMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
}
else
{
Context context = getApplicationContext();
CharSequence text = "Location Not Found!";
int duration = Toast.LENGTH_SHORT;
Toast toast = Toast.makeText(context, text, duration);
toast.show();
}
/*List<Address> addlist=null;
EditText ed=(EditText)findViewById(R.id.txtl);
String loc=ed.getText().toString();
if(loc!=null || !loc.equals(""))
{
```

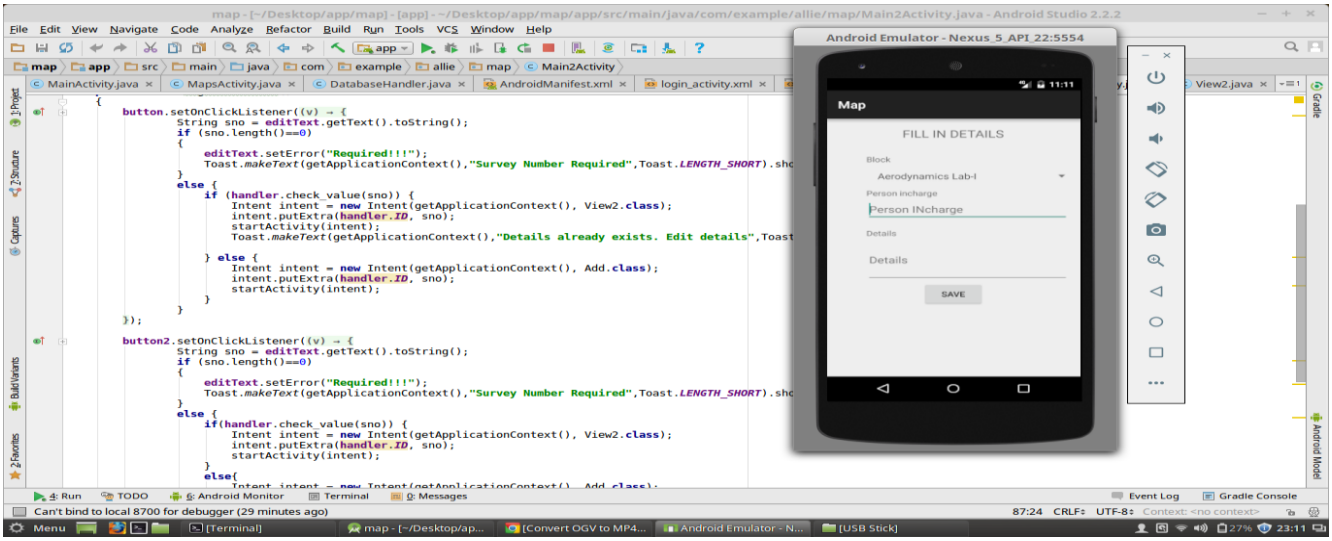
```
Geocoder gc=new Geocoder(this);
try {
addlist=gc.getFromLocationName(loc,1);
} catch (IOException e) {
e.printStackTrace();
}
}
Address add=addlist.get(0);
LatLng latLng=new LatLng(add.getLatitude(),add.getLongitude());
mMap.addMarker(new MarkerOptions().position(latLng).title("Marker"));
mMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));*/


}


public void Ctype(View view)
{
if(mMap.getMapType()==GoogleMap.MAP_TYPE_NORMAL)
{
mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
}
else
{
mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
}
}
```

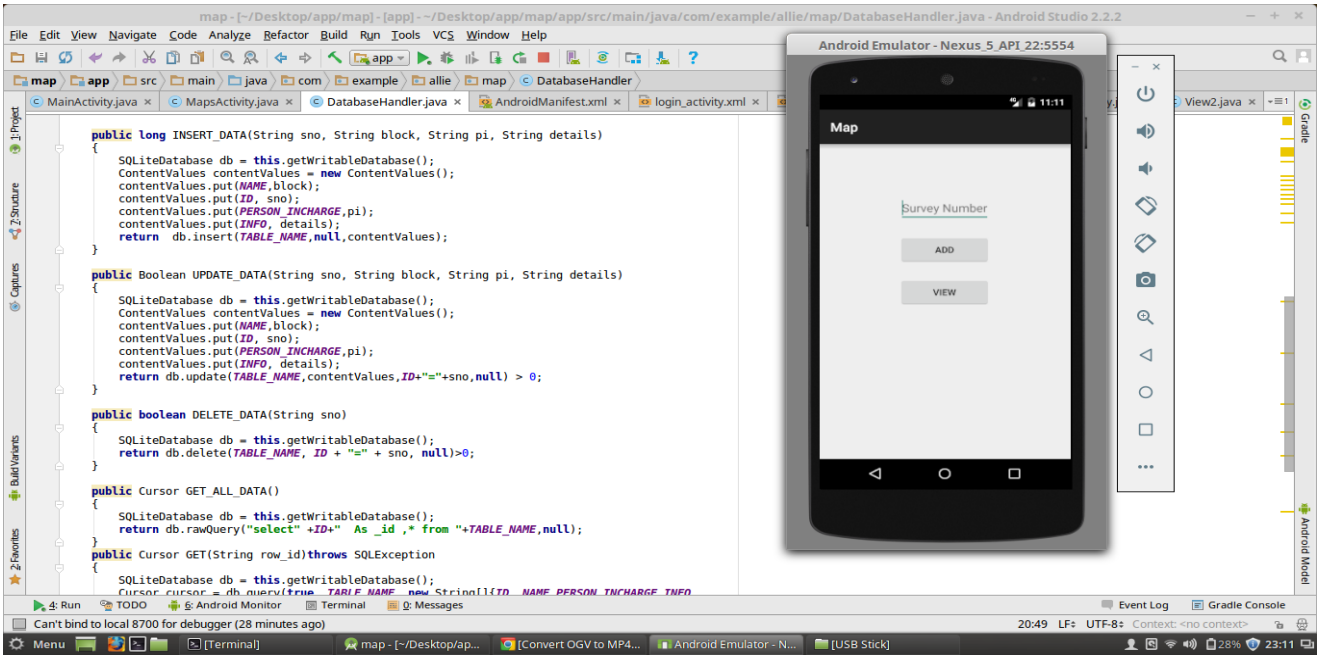
```
private void setUpMapIfNeeded() {
// Do a null check to confirm that we have not already instantiated the map.
if (mMap == null) {
// Try to obtain the map from the SupportMapFragment.
mMap = ((SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map))
.getMap();
// Check if we were successful in obtaining the map.
if (mMap != null) {
setUpMap();
}
}
}
@Override
public void onBackPressed() {
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setMessage("Are you sure you want to logout?")
.setCancelable(false)
.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int id) {
MapsActivity.this.finish();
}
})
.setNegativeButton("No", new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int id) {
```

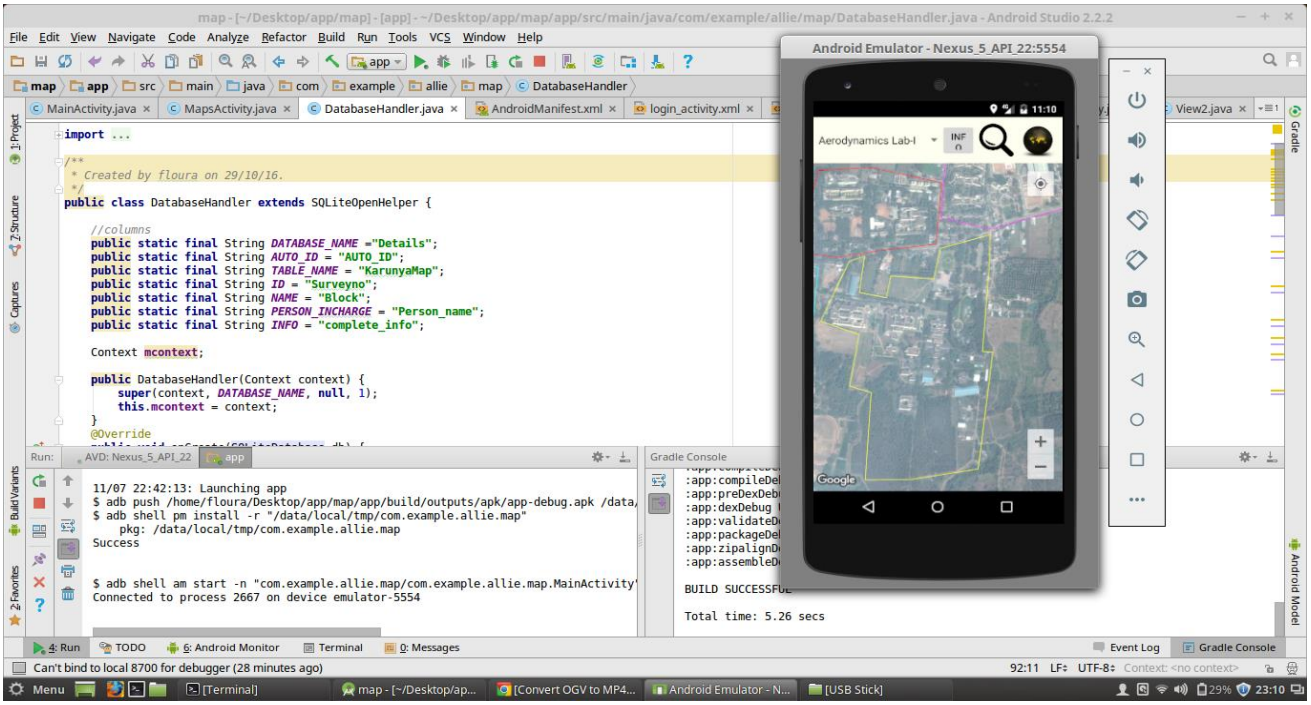
```
dialog.cancel();

}

});

AlertDialog alert = builder.create();

alert.show();

}


private void setUpMap() {

//mMap.addMarker(new MarkerOptions().position(new LatLng(10.93784,
76.74487)).title("Bethesda"));

//mMap.addMarker(new MarkerOptions().position(new LatLng(10.93581,
76.74383)).title("Karunya University"));

mMap.setMyLocationEnabled(true);

mMap.getUiSettings().setZoomControlsEnabled(true);




}



@Override

public void onNothingSelected(AdapterView<?> parent) {


}

}
```

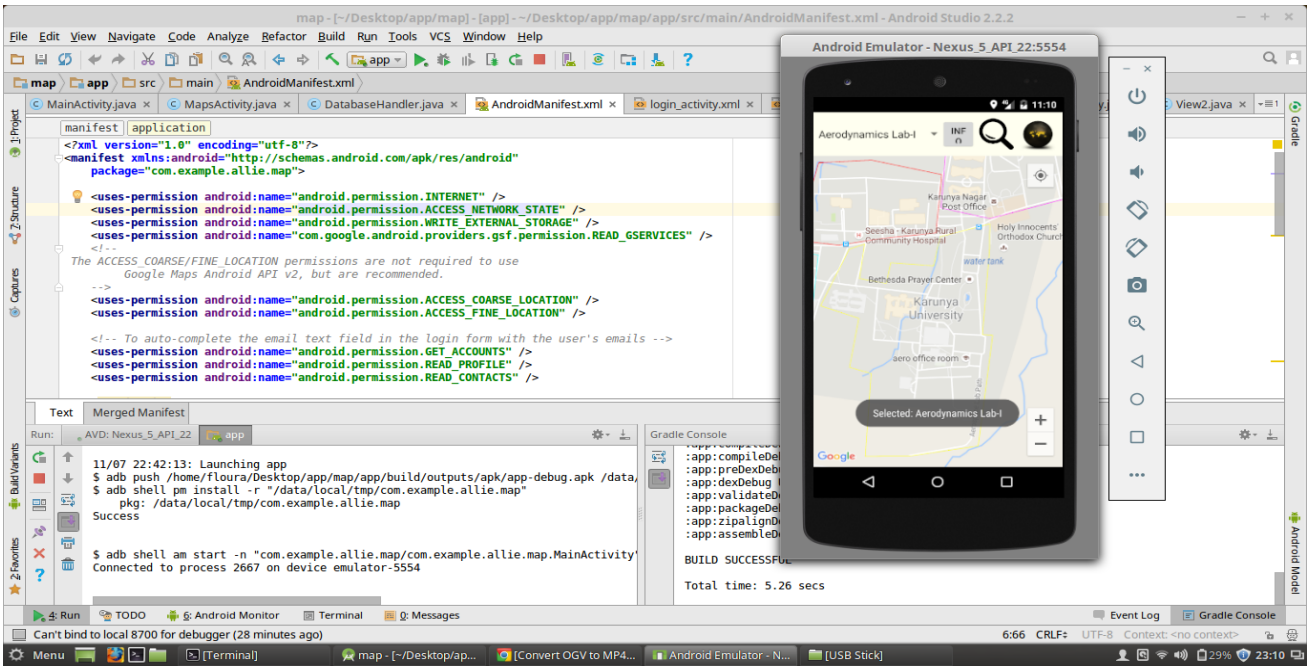
OUTPUT

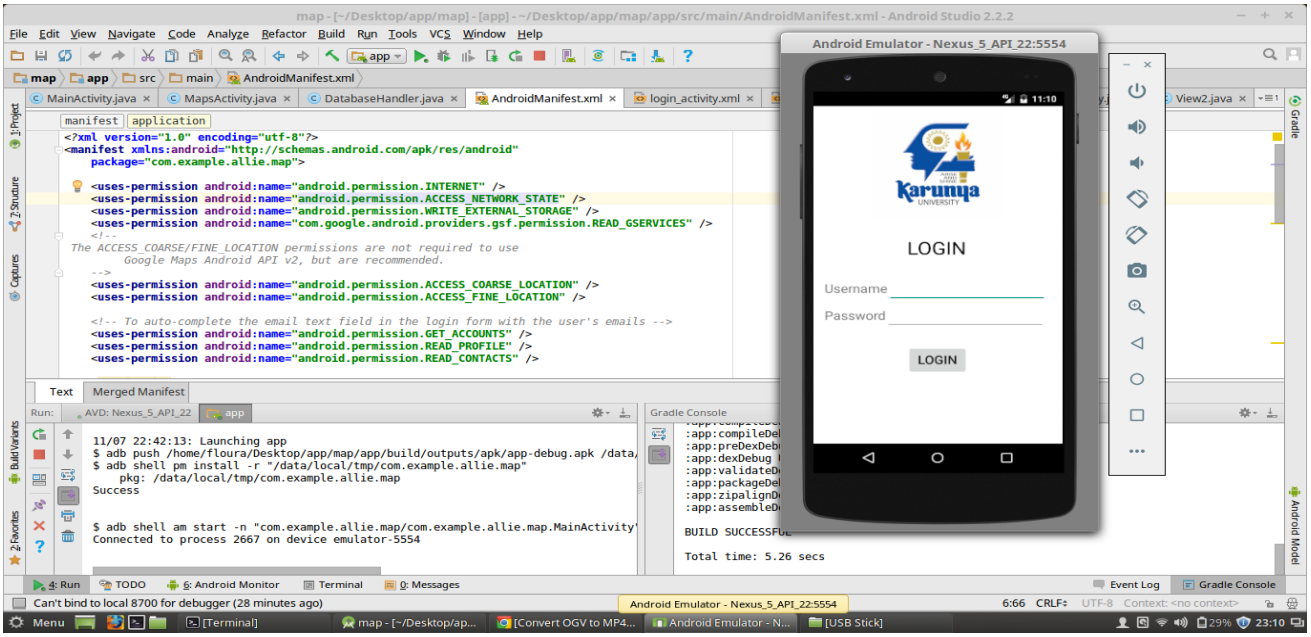

map - [~/Desktop/app/map] - [app] - ~/Desktop/app/map/app/src/main/java/com/example/allie/map/Main2Activity.java - Android Studio 2.2.2
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MainActivity.java
MapsActivity.java
DatabaseHandler.java
AndroidManifest.xml
login_activity.xml
View2.java
button.setOnClickListener((v) → {
String sno = editText.getText().toString();
if (sno.length()==0)
editText.setError("Required!!!");
Toast.makeText(getApplicationContext(),"Survey Number Required",Toast.LENGTH_SHORT).sho
if (handler.check_value(sno)) {
Intent intent = new Intent(getApplicationContext(), View2.class);
intent.putExtra(handler.ID, sno);
startActivity(intent);
Toast.makeText(getApplicationContext(),"Details already exists. Edit details",Toast
} else {
Intent intent = new Intent(getApplicationContext(), Add.class);
intent.putExtra(handler.ID, sno);
startActivity(intent);
button2.setOnClickListener((v) → {
String sno = editText.getText().toString();
if (sno.length()==0)
editText.setError("Required!!!");
Toast.makeText(getApplicationContext(),"Survey Number Required",Toast.LENGTH_SHORT).sho
if(handler.check_value(sno)) {
Intent intent = new Intent(getApplicationContext(), View2.class);
intent.putExtra(handler.ID, sno);
startActivity(intent);
Android Emulator - Nexus_5_API_22:5554
11:11
Map
DETAILS
Block
Aerodynamics Lab-I
Person incharge
sd
Details
SD
sd
as
Aerodynamics Lab-I
4: Run
TODO
6: Android Monitor
Terminal
0: Messages
Event Log
Gradle Console
Can't bind to local 8700 for debugger (29 minutes ago)
87:24 CRLF UTF-8 Context: <no context>
Menu
[Terminal]
map - [~/Desktop/ap...
[Convert OGV to MP4...
Android Emulator - N...
[USB Stick]
23:11

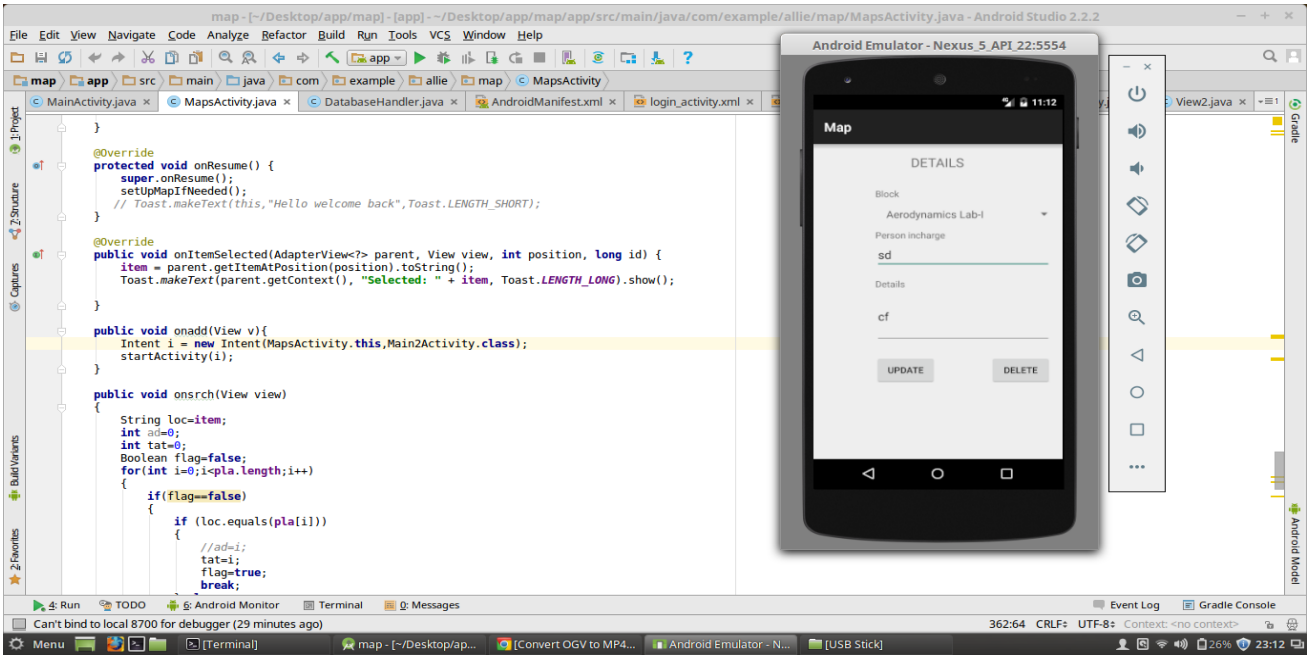

Android Emulator - Nexus_5_API_22:5554
Map
FILL IN DETAILS
Block
Aerodynamics Lab-I
Person incharge
Person INcharge
Details
Details
SAVE


Android Emulator - Nexus_5_API_22:5554
Map
Survey Number
ADD
VIEW


map - [~/Desktop/app/map] - [app] - ~/Desktop/app/map/app/src/main/java/com/example/allie/map/DatabaseHandler.java - Android Studio 2.2.2
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Android Emulator - Nexus_5_API_22:5554
MainActivity.java
MapsActivity.java
DatabaseHandler.java
AndroidManifest.xml
login_activity.xml
import ...
/**
* Created by floura on 29/10/16.
*/
public class DatabaseHandler extends SQLiteOpenHelper {
//columns
public static final String DATABASE_NAME ="Details";
public static final String AUTO_ID = "AUTO_ID";
public static final String TABLE_NAME = "KarunyaMap";
public static final String ID = "Surveyno";
public static final String NAME = "Block";
public static final String PERSON_INCHARGE = "Person_name";
public static final String INFO = "complete_info";
Context mcontext;
public DatabaseHandler(Context context) {
super(context, DATABASE_NAME, null, 1);
this.mcontext = context;
}
@Override
Run: AVD: Nexus_5_API_22 app
11/07 22:42:13: Launching app
$ adb push /home/floura/Desktop/app/map/app/build/outputs/apk/app-debug.apk /data/
$ adb shell pm install -r "/data/local/tmp/com.example.allie.map"
pkg: /data/local/tmp/com.example.allie.map
Success
$ adb shell am start -n "com.example.allie.map/com.example.allie.map.MainActivity"
Connected to process 2667 on device emulator-5554
Gradle Console
BUILD SUCCESSFUL
Total time: 5.26 secs
4: Run TODO 6: Android Monitor Terminal 0: Messages Event Log Gradle Console
Can't bind to local 8700 for debugger (28 minutes ago)
92:11 LF: UTF-8: Context: <no context>
Menu [Terminal] map - [~/Desktop/ap... [Convert OGV to MP4... Android Emulator - N... [USB Stick]
23:10


map - [~/Desktop/app/map] - [app] - ~/Desktop/app/map/app/src/main/AndroidManifest.xml - Android Studio 2.2.2
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Android Emulator - Nexus_5_API_22:5554
MainActivity.java
MapsActivity.java
DatabaseHandler.java
AndroidManifest.xml
login_activity.xml
manifest application
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.allie.map">
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<!--
The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
Google Maps Android API v2, but are recommended.
-->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<!-- To auto-complete the email text field in the login form with the user's emails -->
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.READ_PROFILE" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
Text Merged Manifest
Run: AVD: Nexus_5_API_22 app
11/07 22:42:13: Launching app
$ adb push /home/floura/Desktop/app/map/app/build/outputs/apk/app-debug.apk /data/
$ adb shell pm install -r "/data/local/tmp/com.example.allie.map"
pkg: /data/local/tmp/com.example.allie.map
Success
$ adb shell am start -n "com.example.allie.map/com.example.allie.map.MainActivity"
Connected to process 2667 on device emulator-5554
Gradle Console
BUILD SUCCESSFUL
Total time: 5.26 secs
Selected: Aerodynamics Lab-I
4: Run TODO 6: Android Monitor Terminal 0: Messages Event Log Gradle Console
Can't bind to local 8700 for debugger (28 minutes ago)
6:66 CRLF: UTF-8 Context: <no context>
Menu [Terminal] map - [~/Desktop/ap... [Convert OGV to MP4... Android Emulator - N... [USB Stick]
23:10


Android Emulator - Nexus_5_API_22:5554
LOGIN
Username
Password
LOGIN
Karunya
UNIVERSITY
AndroidManifest.xml
MainActivity.java
MapsActivity.java
DatabaseHandler.java
login_activity.xml
View2.java
Text
Merged Manifest
Run:
Gradle Console
Total time: 5.26 secs


Android Emulator - Nexus_5_API_22:5554
Map
DETAILS
Block
Aerodynamics Lab-I
Person incharge
sd
Details
cf
UPDATE
DELETE
MapsActivity.java
MainActivity.java
DatabaseHandler.java
AndroidManifest.xml
login_activity.xml
View2.java