Military Database management system (Soldier Database)

Siddharth Das - 18BIT0379
Database Management and Systems – ITE1003
VIT University, Vellore (November, 2019)
Department Of Information Technology
Vellore, Tamil Nadu
India

<u>Abstract:</u>

In military bases, the basic data like information of soldiers, their medical records, family details, their past trainings, skills and weapons they are familiar with are to be stored. There are many classes of soldiers and a large range or training, skills and arms used.

REVIEW 1:

Data Collection and Entity Sets Identification

<u>Entity sets:</u>

Soldier(s_id, s_firstn, s_lastn, s_DOB, s_age, s_gen, s_rank, s_posting, s_cor, s_state, s_phone,s_email, ismarried)
This entity set is used to store details about the soldier. s_id is the primary key.
In this entity set : The attribute s_phone is multivalued. The attribute s_age is derived from s_DOB.The rest of the attributes are all simple and single valued

Father(f-id, f_firstn, f_lastn, f_DOB, f_phone, f_email , f_age, f_city, f_state, f_adline)
This is to record information related to the soldier's father. f_id is the primary key for this table.
In this entity set: The attribute f_phone is multivalued. The attribute f_age is derived from f_DOB. The rest of the attributes are all simple and single valued.

Mother(m-id, m_firstn, m_lastn, m_DOB, m_phone, m_email , m_age, m_city, m_state, m_adline)
This is to record information related to the soldier's mother. m_id is the primary key for this table.
In this entity set: The attribute m_phone is multivalued. The attribute m_age is derived from m_DOB. The rest of the attributes are all simple and single valued.

Spouse(sp_id, sp_firstn, sp_lastn, sp_DOB, sp_phone, sp_email , sp_age, sp_gen, sp_city, sp_state, sp_adline)
This is to record information related to the soldier's spouse. sp_id is the primary key for this table.
In this entity set: The attribute sp_phone is multivalued. The attribute sp_age is derived from sp_DOB. The rest of the attributes are all simple and single valued.

Medical(m_id, bg, height, weight, hearing, isdiabetic)
This table is to record the soldier's basic medical data.m_id is the primary key for this table.
In this entity set all the attributes are simple and single valued.

Vaccine(v_id, polio, tetanus, DDT, HIV, POX)
The list of vaccines that the soldier has taken is stored in this table. The primary key is v_id.
In this entity set all the attributes are simple and single valued.

Skillset(sk_id, sprint_speed, climb_speed, computing, teamwork)
This table stores the skillsets of the soldier. The primary key is sk_id.
In this entity set all the attributes are simple and single valued.

Arms_used(a_id, INAS, AK103, MG56, AKM, Dragunov_svd59) Arms used by the soldier are stored in this table. a_id is the primary key. In this entity set all the attributes are simple and single valued.

Bank(ifsc, b_name, b_branch)
This table stores the bank details of the soldier. ifsc is the primary key.
In this entity set all the attributes are simple and single valued.

<u>Relationship sets:</u>

Relationship sets	Descriptive attributes	Description about relationship and its type
Family		Each soldier will have will have one family consisting of a father, mother and spouse(if the soldier is married) and each family will have one soldier. So, the relationship from Soldier to Father, Mother and Spouse is One-to-One relationship.
Health		One soldier will have one medical record and one medical record can belong to one soldier only. Thus Soldier to medical is a One-to-One relationship.
MV		One medical report may have one detailed report for vaccines and one vaccine report may belong to one medical report only. Thus the relationship between Medical and Vaccine is One-to-One relationship.
ME		One medical report may have one detailed report for vision and one vision report may belong to one medical report only. Thus the relationship between Medical and Vision is One-to-One relationship.
Trained		One soldier can have only one skill set but one skill set can belong to many soldiers. So, the relationship between Soldier and skillset is Many-to-One relationship.
SA		One soldier can have only one record for arms used but one record of arms used can belong to many soldiers. So, the relationship between Soldier and Arms_used is Many-to-One relationship.
SB		One soldier can have bank accounts one bank can provide accounts to many soldiers. So, the relationship between Soldier and Bank is Many-to-Many relationship.

The data requirements & the necessary integrity constraints that are reasonable for the database under consideration in the above tables.

The functional requirements should involve different scenarios:

- Removal of old data
- Modification of existing data
- Data Retrieval

REMOVAL OF OLD DATA:

Case 1: If the soldier is retired, the maximum percentage of data regarding the present usage and stats will be removed and only the basic data of the military achievements, rank and identity is kept for future references.

Case 2: If the soldier is martyred in any war or rebellion, the official records essential for present military usage is periodically removed and the past duty records, rank and achievements are stored in the database respectively.

MODIFICATION OF EXISTING DATA:

If the soldier's field of duty is transferred to a different military regiment or posted or stationed to different battalion, the data is modified accordingly.

If the rank of the soldier is General, he is assigned with the AK103 weapon

DATA RETRIEVAL:

The data and the informational functionalities can be retrieved from the military database, when required. It stores the essential data for future references. For Example:

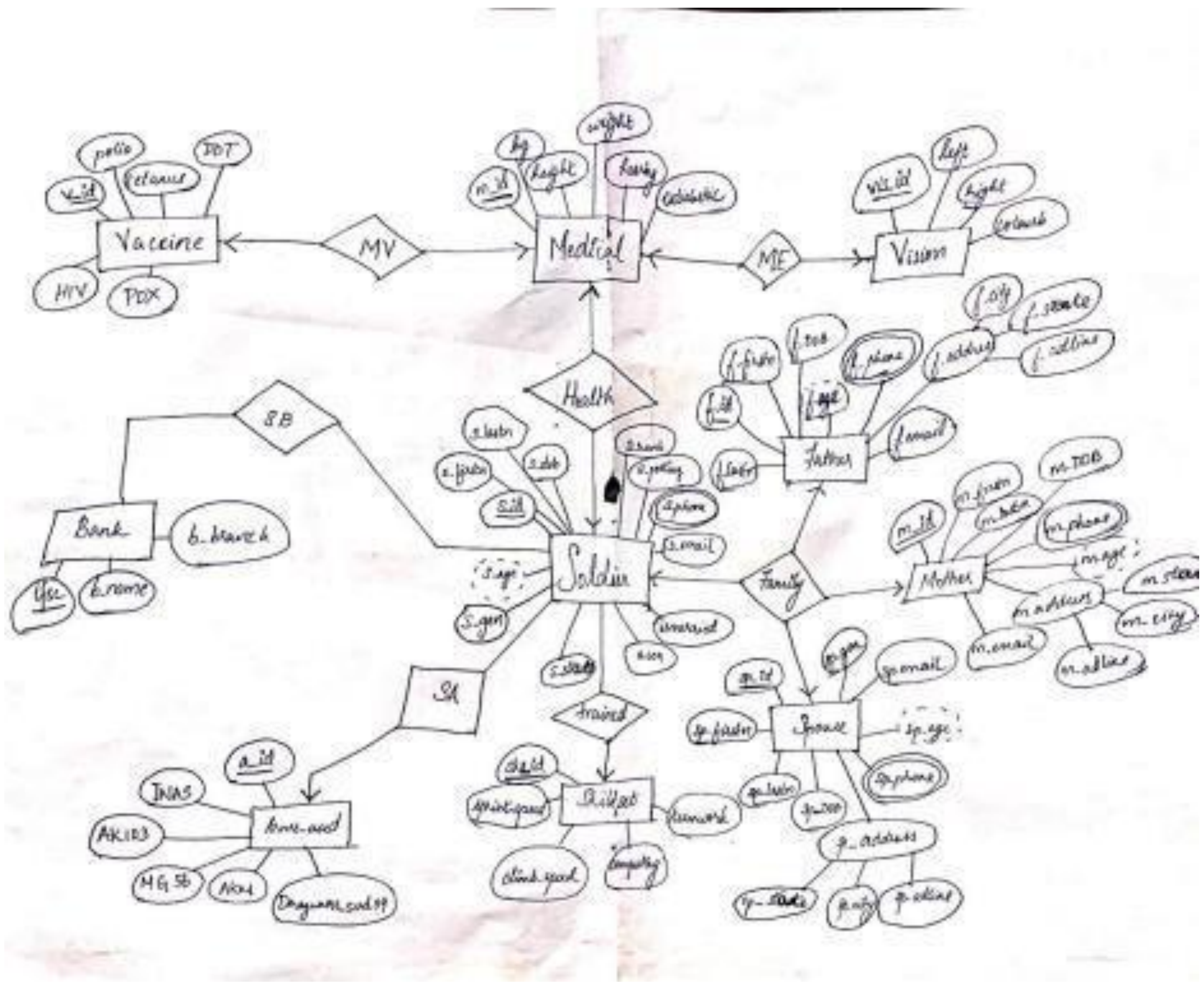
1 – The number of soldiers posted at a particular posting area. For example, Lansdowne.

2 – To find the number of soldiers carrying a particular weapon.

3 - Regular check-ups of the soldier have to be done and compared with the previous results of the soldier for which the medical history has to be retrieved from the database.

4 - When soldiers are paid, the bank details of the soldiers must be retrieved with the soldier’s details, in descending order of the IFSC code.

ER Diagram


Vaccine
v_id
polio
tetanus
DPT
HIV
POX
MV
Medical
m_id
bg
height
weight
diabetic
ME
Vision
vis_id
left
right
Health
SB
Bank
b_branch
ifsc
b_name
Soldier
s_dob
s_age
s_gen
s_phone
s_email
unmarried
s_state
SA
Arms_used
a_id
INAS
AK103
MG 5b
Dragunov_svd 99
trained
Skillset
sprint_speed
climb_speed
teamwork
Family
Father
f_id
f_dob
f_age
f_phone
f_address
f_city
f_state
f_addline
f_email
Mother
m_id
m_DOB
m_phone
m_age
m_address
m_state
m_city
m_addline
m_email
Spouse
sp_id
sp_gen
sp_email
sp_age
sp_phone
sp_DOB
sp_address
sp_state
sp_city
sp_addline

Data Representation:

Table names and attribute names should be appropriate and meaningful.

Each attribute must be defined with appropriate data type, size and one or more of the

following constraints;

o NOT NULL

o UNIQUE

o PRIMARY KEY

o FOREIGN KEY

o CHECK

Each table must be inserted with few valid records. Also, generate questions that involve SELECT, FROM and WHERE clauses and answer those questions with appropriate SQL queries. At least 30 questions and queries should be generated. Use queries that shows in result all attributes, chosen attributes, arithmetic calculation in SELECT clause, multiple conditions connected with logical connectives etc.

REVIEW 2:

Relational Database

Reduction of ER to schema:

1. Soldier(s_id, s_firstn, s_lastn, s_DOB, s_state, s_cor, s_phone, s_email, ismarried, s_rank, s_posting, sk_id, a_id)
2. Father(f_id, f_firstn, f_lastn, f_DOB, ,f_city, f_state, f_adline, f_phone, f_email, s_id)
3. Mother(m_id, m_firstn, m_lastn, m_DOB, ,m_city, m_state, m_adline, m_phone, m_email, s_id)
4. Spouse(sp_id, sp_firstn, sp_lastn, sp_DOB, ,sp_city, sp_state, sp_adline, sp_phone, sp_email, s_id)
5. Medical(m_id, bg, height, weight, hearing, isdiabetic, s_id, v_id, vis_id)
6. Vaccine(v_id, polio, tetanus, DDT, HIV, POX)
7. Vision(vis_id, left, right, colourb)
8. Skillset(sk_id, sprint_speed, climb_speed, computing, teamwork)
9. Arms_used(a_id, INAS, AK103, MG56, AKM, Dragunov_svd59)
10. Bank(ifsc, b_name, b_branch)
11. SB(ifsc, s_id)

1. Soldier: The relationship is one to one from Soldier to father.so, the primary key of Soldier is foreign key of Father.

2. Father: The relationship is one to one from Father to soldier. So, the primary key of Soldier is the foreign Key of father.

3. Mother: The relationship is One to one from Mother to Soldier, So, the primary key of mother is the foreign key of Soldier.

4. Spouse: The relationship is one to one from Spouse to Soldier. So, the primary key of spouse is the foreign key of soldier.

5. Medical: The relationship is one to one from Medical to Soldier. So, the primary key of Medical is the foreign key of Soldier.

6. Vaccine: The relationship is one to one from Vaccine to medical. So, the primary key of Vaccine is the foreign key of medical.

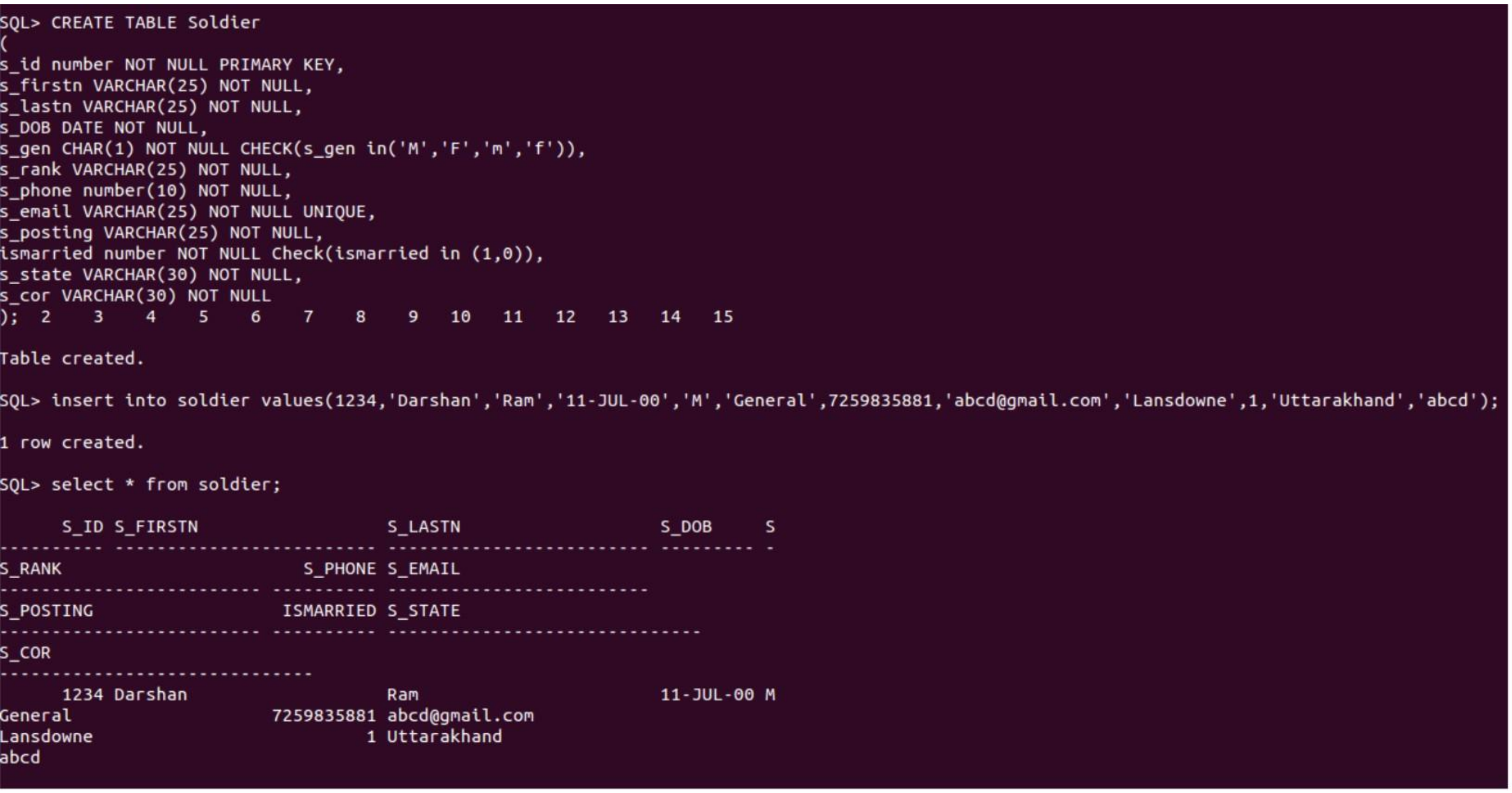
7. Vision: The relationship is one to one from Vision to Medical. So, the primary key of Vision is the foreign key of Medical.

8. Skillset: The relationship is one to many from skillset to soldier.so the primary key of skillset is the foreign key in soldier.

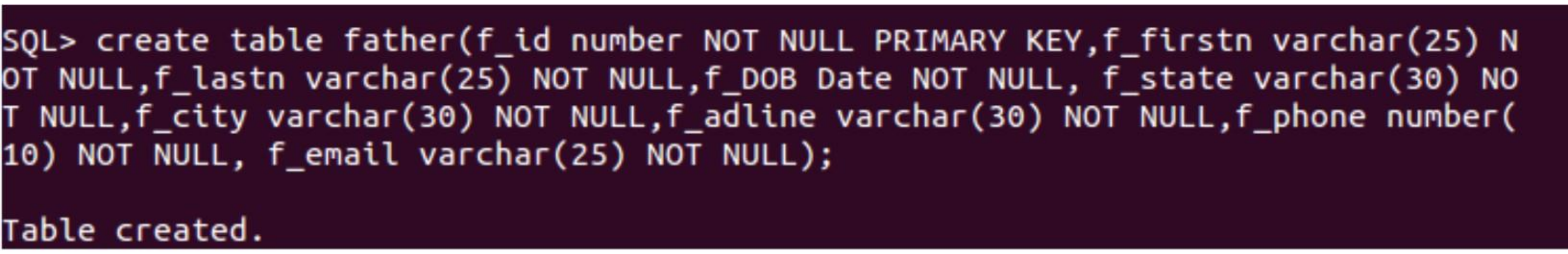
9. arms used: The relationship is one to many from arms used to soldier. So the primary key of arms used is the foreign key of the soldiers.

10. Bank: The relationship is many to many from bank to soldiers.so the primary key of bank is the foreign key in soldiers.

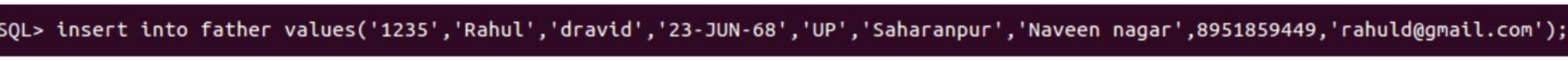
11. SB: The relationship is many to many from SB to soldiers.so the primary key of SB is used as the foreign key of soldiers.



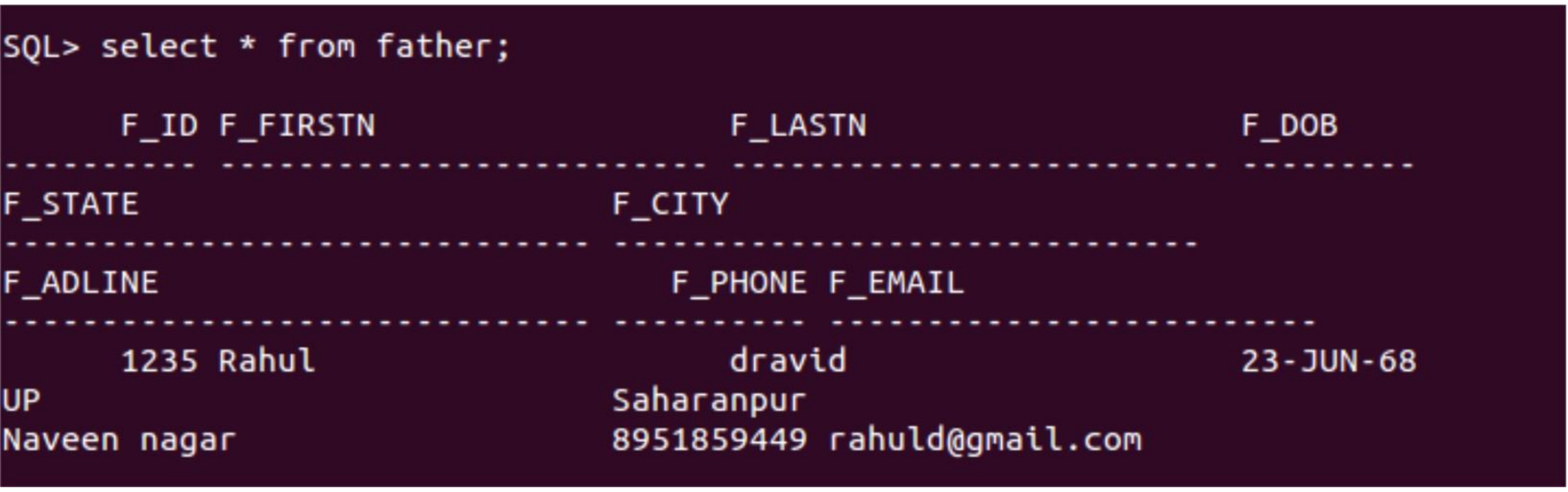
```
SQL> CREATE TABLE Soldier
(
s_id number NOT NULL PRIMARY KEY,
s_firstn VARCHAR(25) NOT NULL,
s_lastn VARCHAR(25) NOT NULL,
s_DOB DATE NOT NULL,
s_gen CHAR(1) NOT NULL CHECK(s_gen in('M','F','m','f')),
s_rank VARCHAR(25) NOT NULL,
s_phone number(10) NOT NULL,
s_email VARCHAR(25) NOT NULL UNIQUE,
s_posting VARCHAR(25) NOT NULL,
ismarried number NOT NULL Check(ismarried in (1,0)),
s_state VARCHAR(30) NOT NULL,
s_cor VARCHAR(30) NOT NULL
);  2    3    4    5    6    7    8    9   10   11   12   13   14   15

Table created.

SQL> insert into soldier values(1234,'Darshan','Ram','11-JUL-00','M','General',7259835881,'abcd@gmail.com','Lansdowne',1,'Uttarakhand','abcd');

1 row created.

SQL> select * from soldier;

      S_ID S_FIRSTN                  S_LASTN                   S_DOB     S
---------- ------------------------- ------------------------- --------- -
S_RANK                       S_PHONE S_EMAIL
------------------------- ---------- -------------------------
S_POSTING                  ISMARRIED S_STATE
------------------------- ---------- ------------------------------
S_COR
------------------------------
      1234 Darshan                   Ram                       11-JUL-00 M
General                   7259835881 abcd@gmail.com
Lansdowne                          1 Uttarakhand
abcd
```



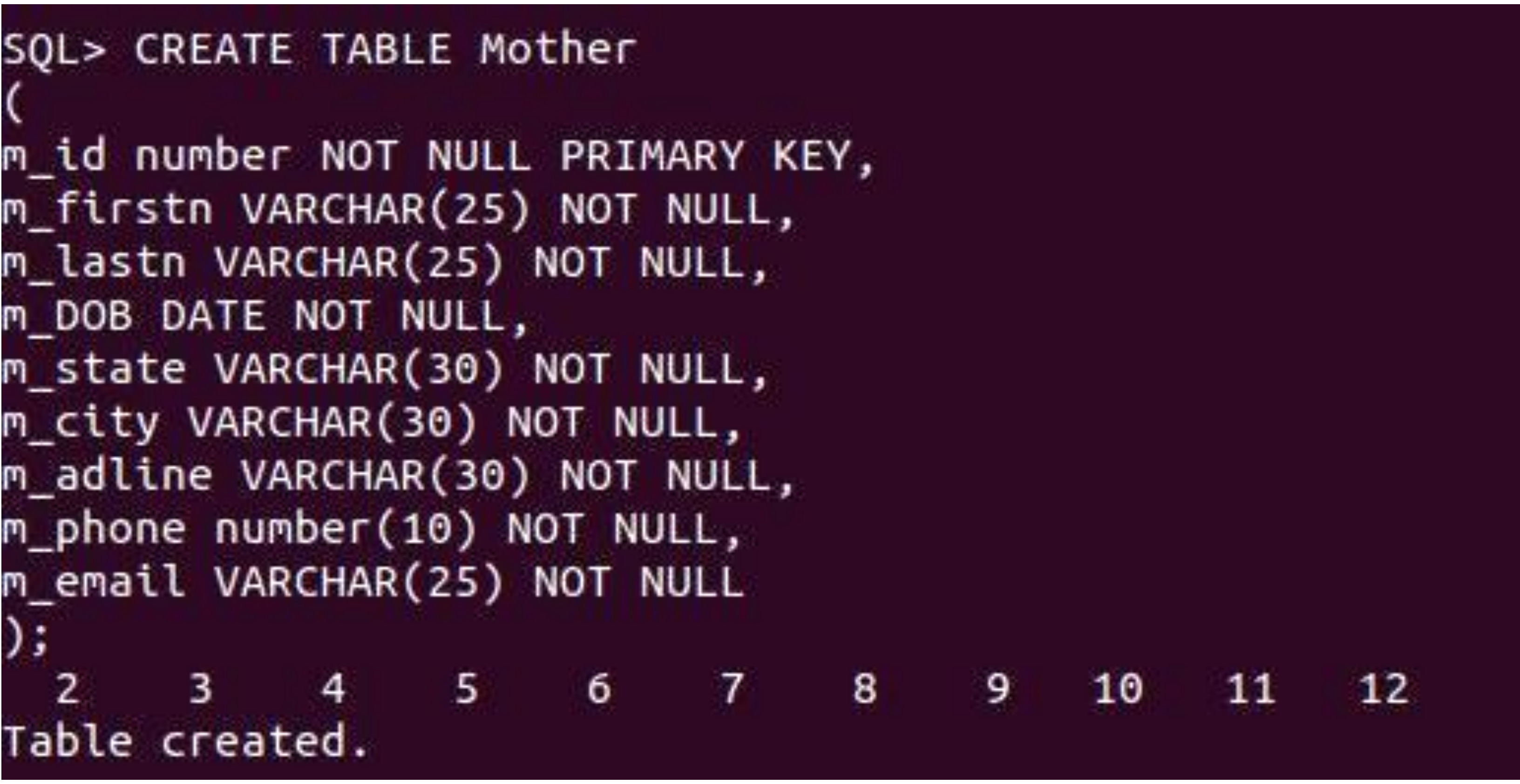
```
SQL> create table father(f_id number NOT NULL PRIMARY KEY,f_firstn varchar(25) N
OT NULL,f_lastn varchar(25) NOT NULL,f_DOB Date NOT NULL, f_state varchar(30) NO
T NULL,f_city varchar(30) NOT NULL,f_adline varchar(30) NOT NULL,f_phone number(
10) NOT NULL, f_email varchar(25) NOT NULL);

Table created.
```



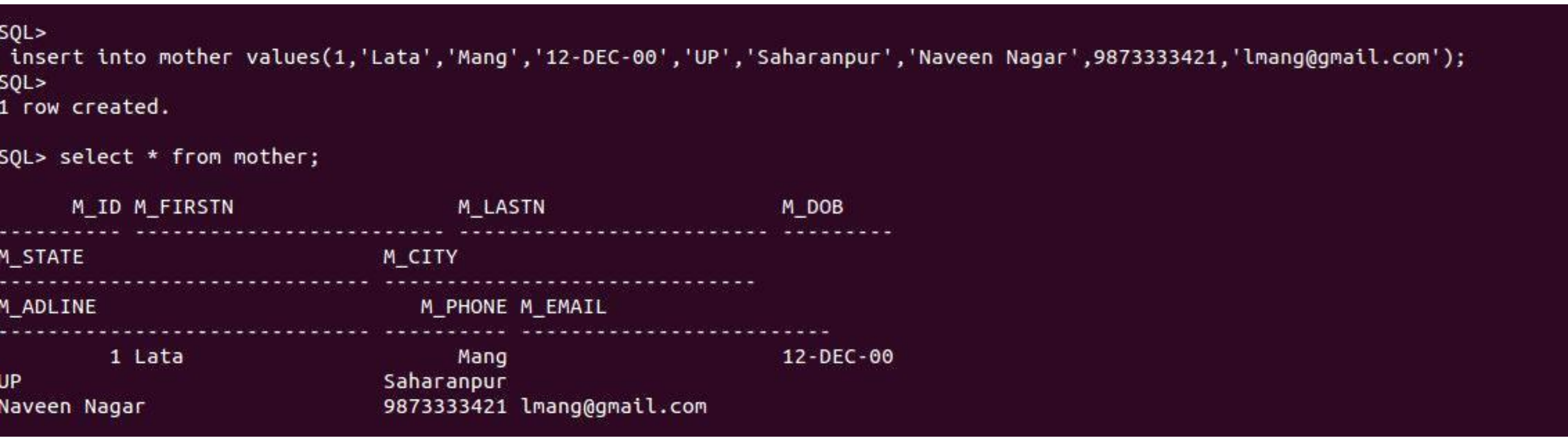
```
SQL> insert into father values('1235','Rahul','dravid','23-JUN-68','UP','Saharanpur','Naveen nagar',8951859449,'rahuld@gmail.com');
```



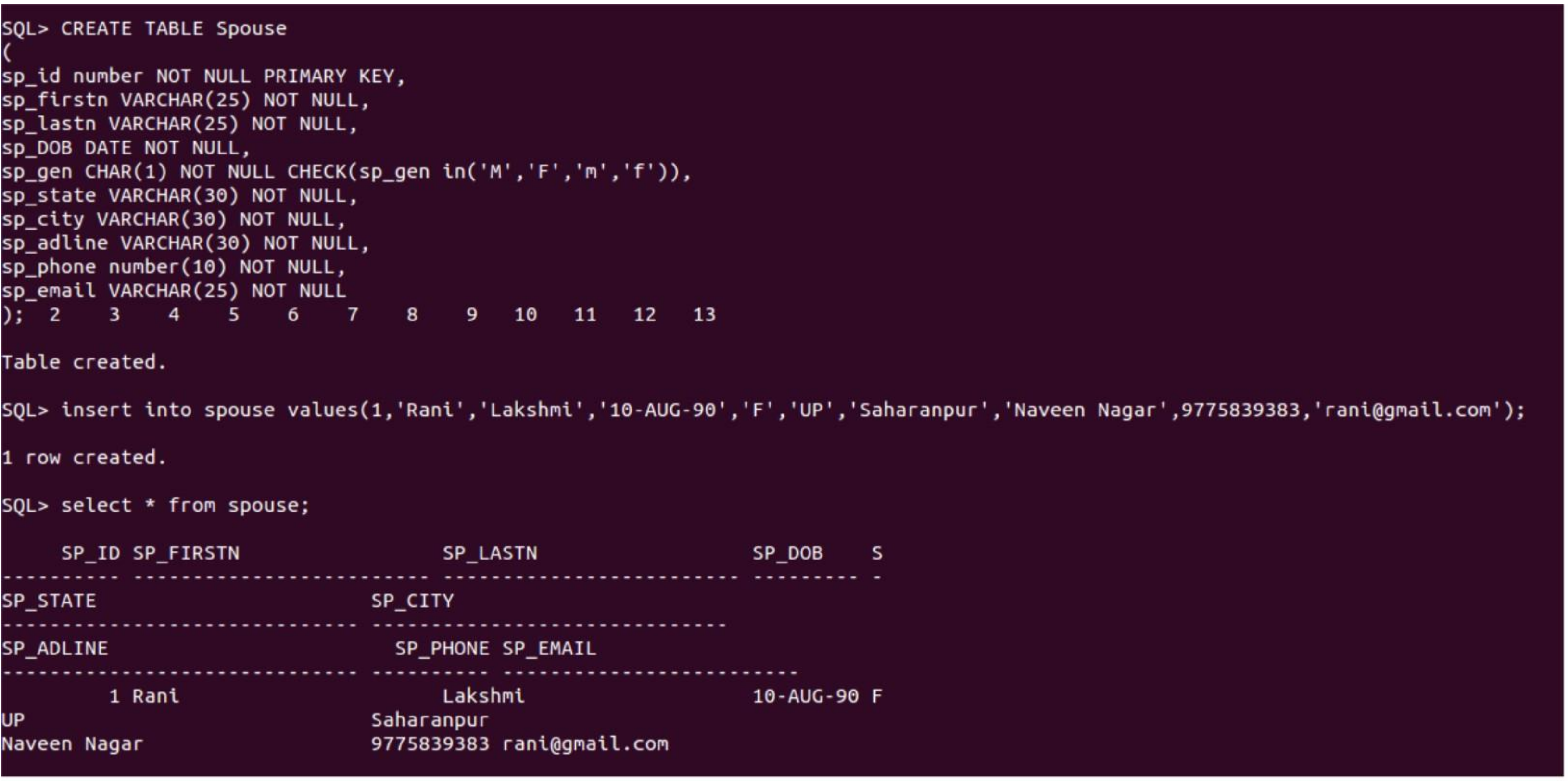
```
SQL> select * from father;

      F_ID F_FIRSTN                  F_LASTN                   F_DOB
---------- ------------------------- ------------------------- ---------
F_STATE                        F_CITY
------------------------------ ------------------------------
F_ADLINE                          F_PHONE F_EMAIL
------------------------------ ---------- -------------------------
      1235 Rahul                     dravid                    23-JUN-68
UP                             Saharanpur
Naveen nagar                   8951859449 rahuld@gmail.com
```



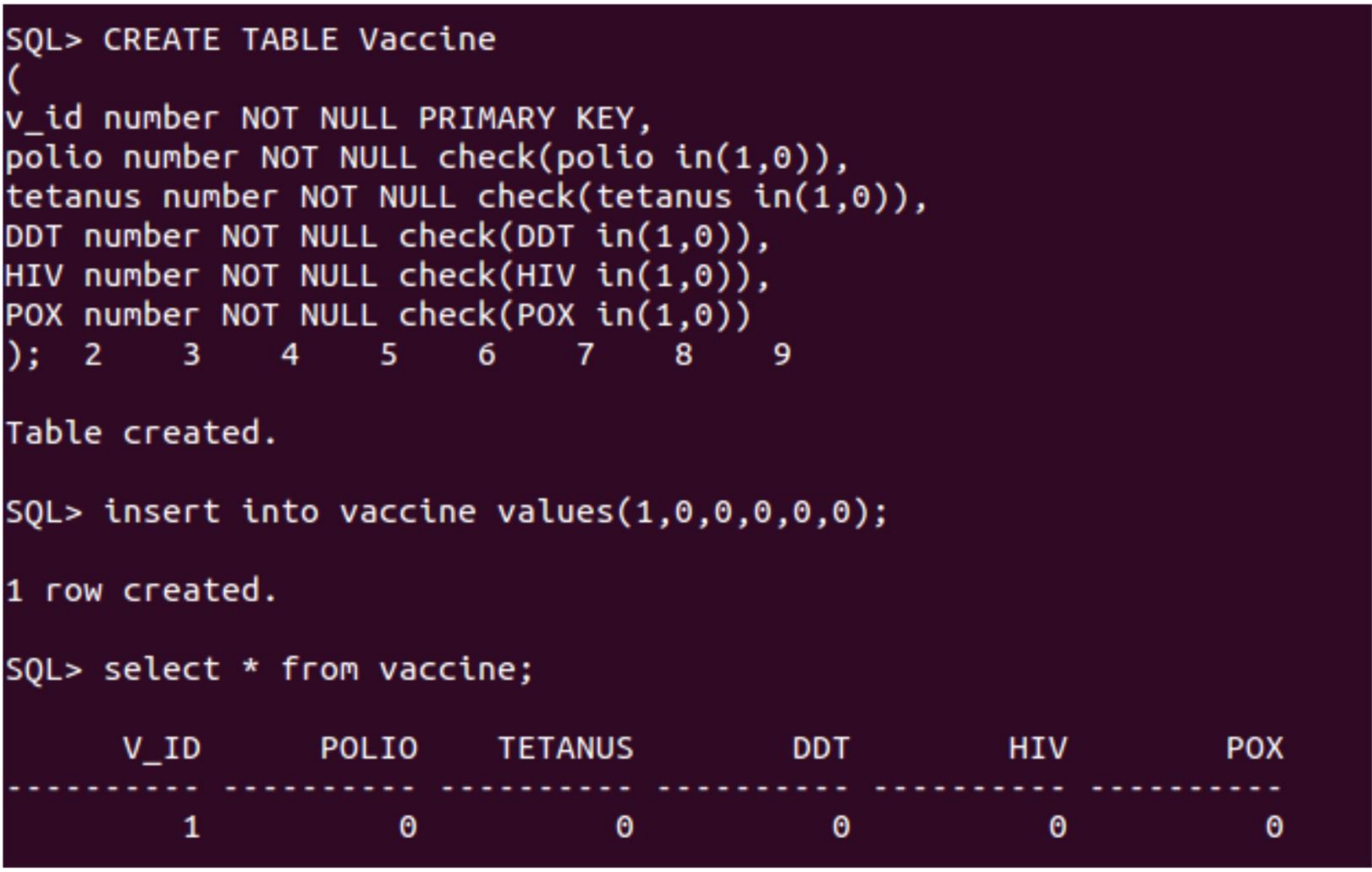
```
SQL> CREATE TABLE Mother
(
m_id number NOT NULL PRIMARY KEY,
m_firstn VARCHAR(25) NOT NULL,
m_lastn VARCHAR(25) NOT NULL,
m_DOB DATE NOT NULL,
m_state VARCHAR(30) NOT NULL,
m_city VARCHAR(30) NOT NULL,
m_adline VARCHAR(30) NOT NULL,
m_phone number(10) NOT NULL,
m_email VARCHAR(25) NOT NULL
);
Table created.
```



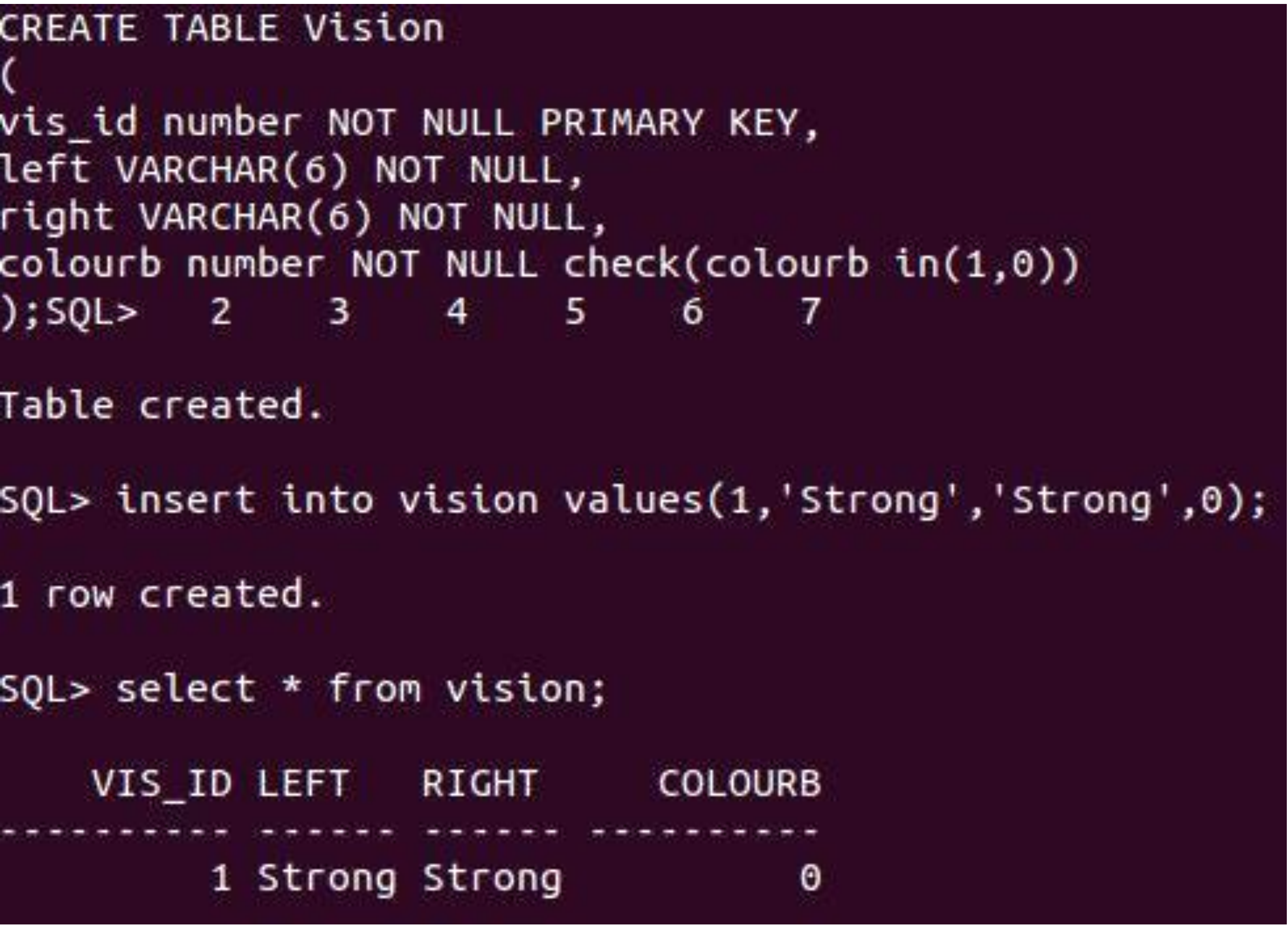
```
SQL>
 insert into mother values(1,'Lata','Mang','12-DEC-00','UP','Saharanpur','Naveen Nagar',9873333421,'lmang@gmail.com');
SQL>
1 row created.

SQL> select * from mother;

      M_ID M_FIRSTN                  M_LASTN                   M_DOB
---------- ------------------------- ------------------------- ---------
M_STATE                        M_CITY
------------------------------ ------------------------------
M_ADLINE                          M_PHONE M_EMAIL
------------------------------ ---------- -------------------------
         1 Lata                      Mang                      12-DEC-00
UP                             Saharanpur
Naveen Nagar                   9873333421 lmang@gmail.com
```



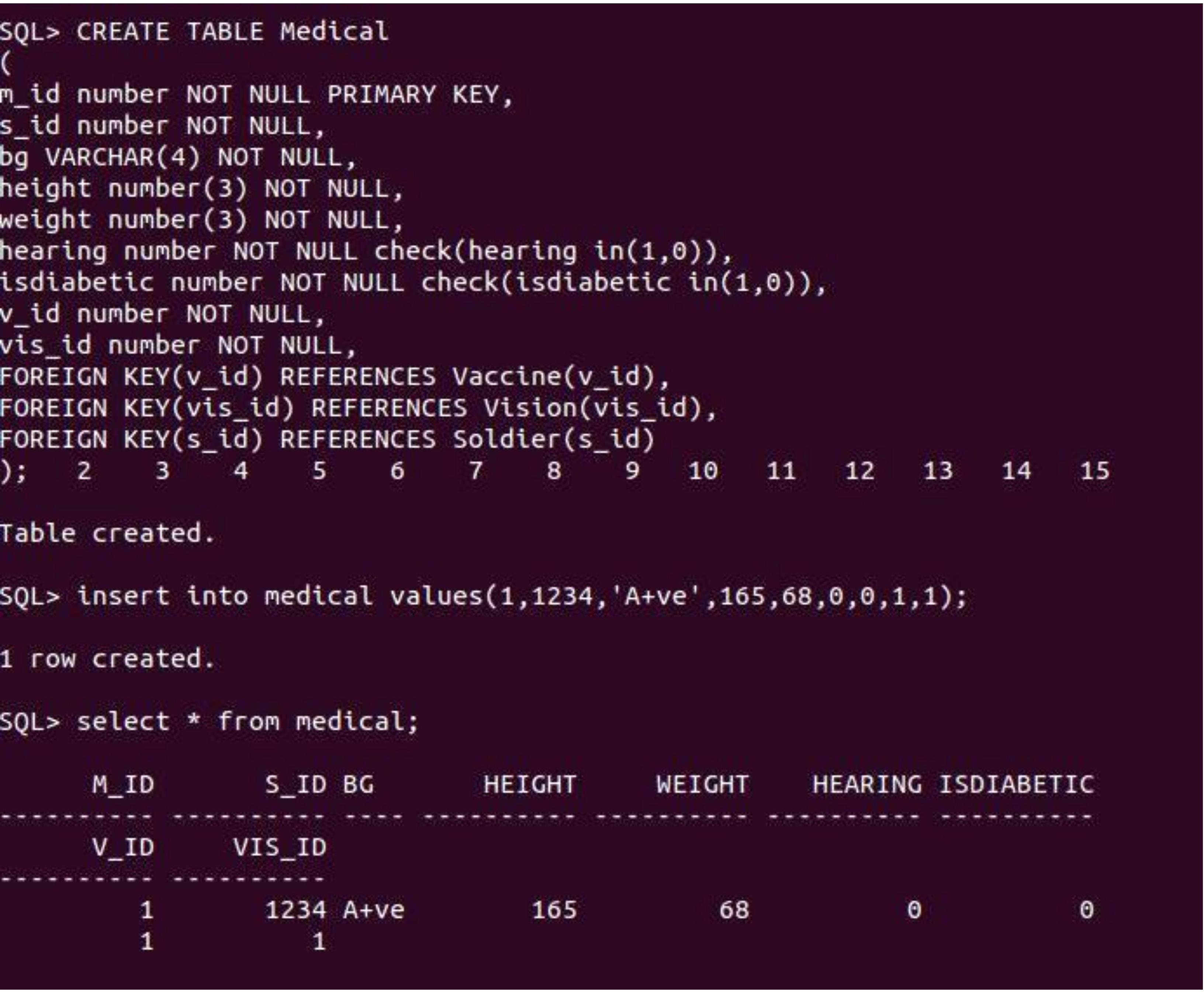
```
SQL> CREATE TABLE Spouse
(
sp_id number NOT NULL PRIMARY KEY,
sp_firstn VARCHAR(25) NOT NULL,
sp_lastn VARCHAR(25) NOT NULL,
sp_DOB DATE NOT NULL,
sp_gen CHAR(1) NOT NULL CHECK(sp_gen in('M','F','m','f')),
sp_state VARCHAR(30) NOT NULL,
sp_city VARCHAR(30) NOT NULL,
sp_adline VARCHAR(30) NOT NULL,
sp_phone number(10) NOT NULL,
sp_email VARCHAR(25) NOT NULL
);  2    3    4    5    6    7    8    9   10   11   12   13

Table created.

SQL> insert into spouse values(1,'Rani','Lakshmi','10-AUG-90','F','UP','Saharanpur','Naveen Nagar',9775839383,'rani@gmail.com');

1 row created.

SQL> select * from spouse;

     SP_ID SP_FIRSTN                 SP_LASTN                  SP_DOB    S
---------- ------------------------- ------------------------- --------- -
SP_STATE                       SP_CITY
------------------------------ ------------------------------
SP_ADLINE                        SP_PHONE SP_EMAIL
------------------------------ ---------- -------------------------
         1 Rani                      Lakshmi                   10-AUG-90 F
UP                             Saharanpur
Naveen Nagar                   9775839383 rani@gmail.com
```



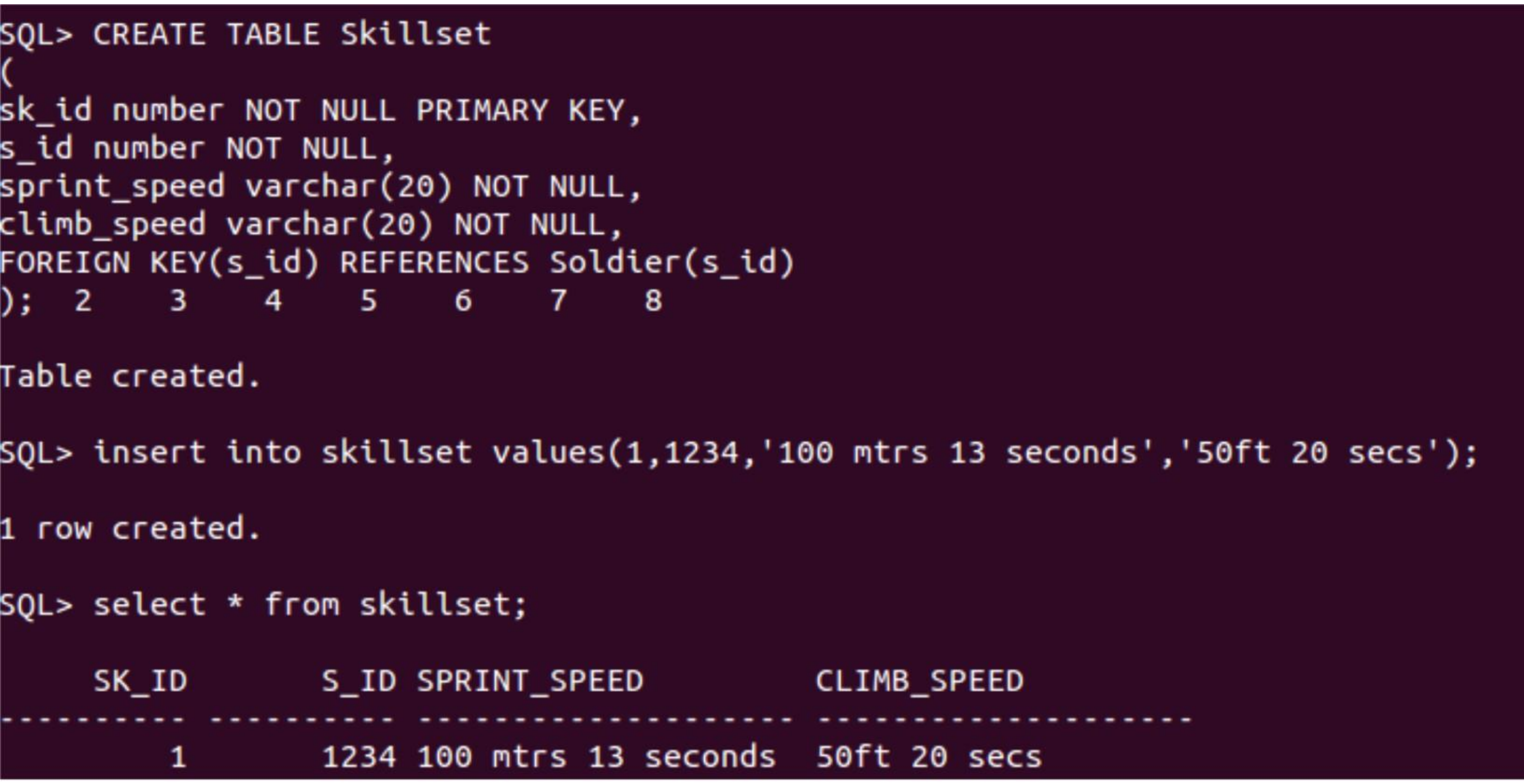
```
SQL> CREATE TABLE Vaccine
(
v_id number NOT NULL PRIMARY KEY,
polio number NOT NULL check(polio in(1,0)),
tetanus number NOT NULL check(tetanus in(1,0)),
DDT number NOT NULL check(DDT in(1,0)),
HIV number NOT NULL check(HIV in(1,0)),
POX number NOT NULL check(POX in(1,0))
);  2    3    4    5    6    7    8    9

Table created.

SQL> insert into vaccine values(1,0,0,0,0,0);

1 row created.

SQL> select * from vaccine;

      V_ID      POLIO    TETANUS        DDT        HIV        POX
---------- ---------- ---------- ---------- ---------- ----------
         1          0          0          0          0          0
```



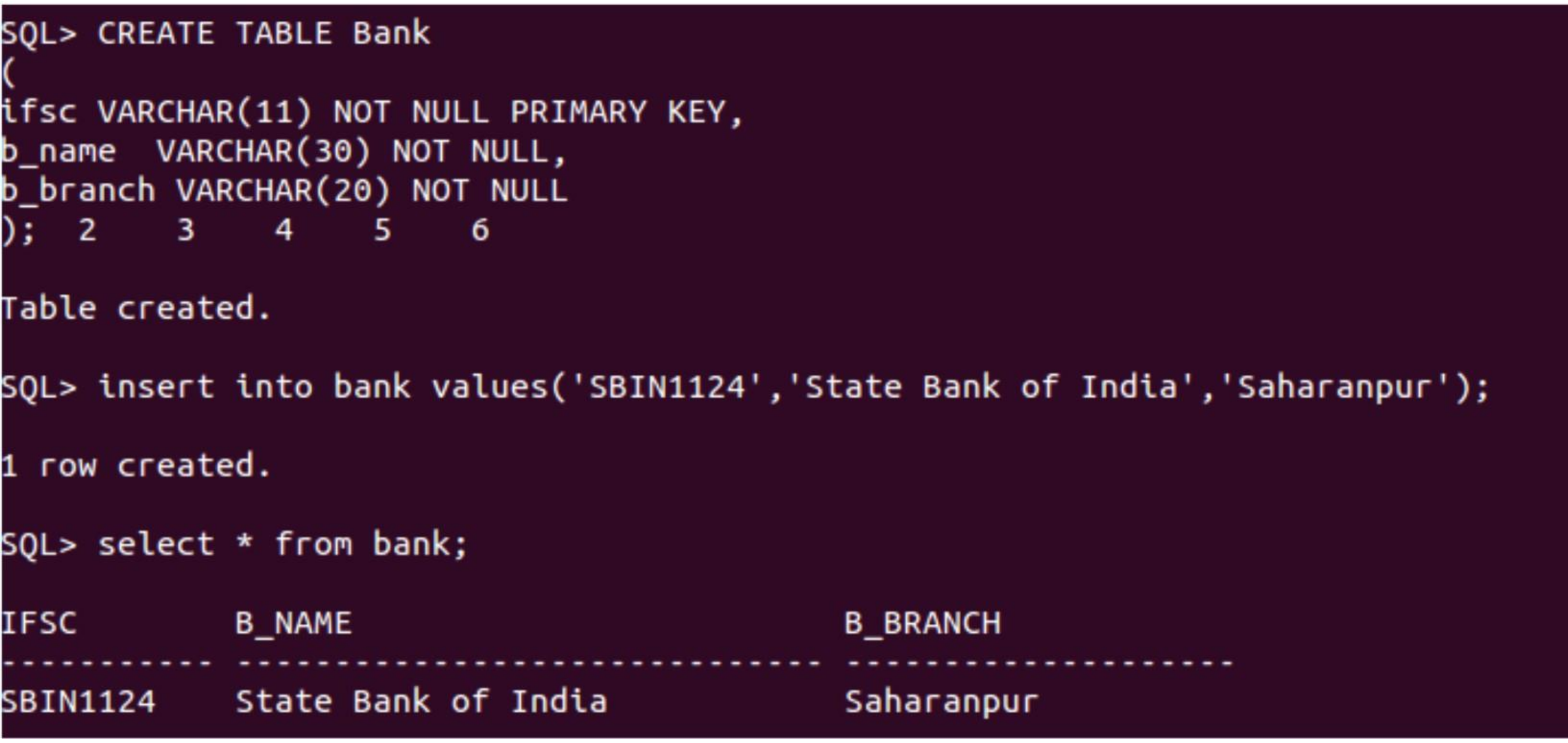
```
CREATE TABLE Vision
(
vis_id number NOT NULL PRIMARY KEY,
left VARCHAR(6) NOT NULL,
right VARCHAR(6) NOT NULL,
colourb number NOT NULL check(colourb in(1,0))
);SQL>   2    3    4    5    6    7

Table created.

SQL> insert into vision values(1,'Strong','Strong',0);

1 row created.

SQL> select * from vision;

    VIS_ID LEFT   RIGHT     COLOURB
---------- ------ ------ ----------
         1 Strong Strong          0
```



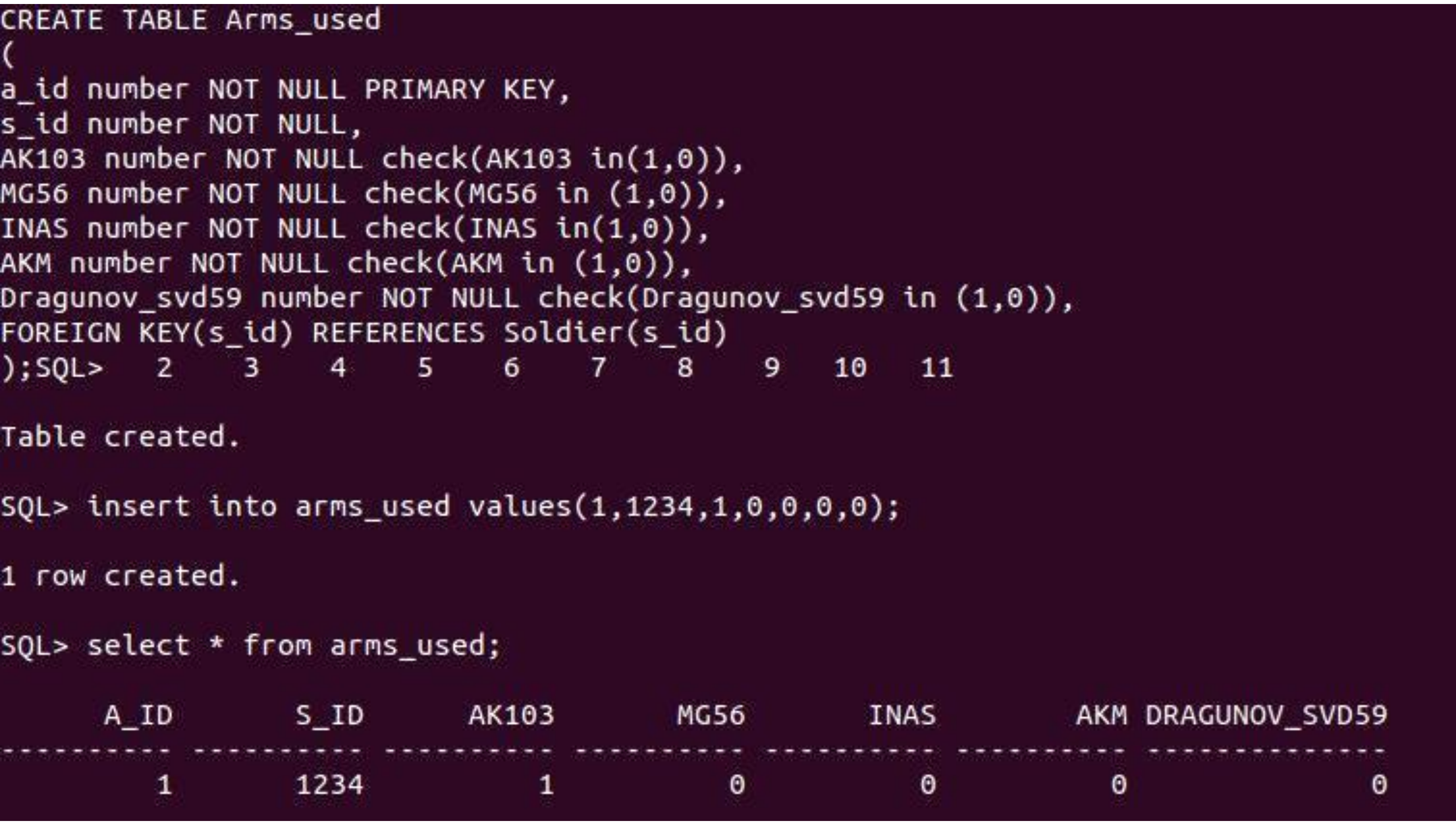
```
SQL> CREATE TABLE Medical
(
m_id number NOT NULL PRIMARY KEY,
s_id number NOT NULL,
bg VARCHAR(4) NOT NULL,
height number(3) NOT NULL,
weight number(3) NOT NULL,
hearing number NOT NULL check(hearing in(1,0)),
isdiabetic number NOT NULL check(isdiabetic in(1,0)),
v_id number NOT NULL,
vis_id number NOT NULL,
FOREIGN KEY(v_id) REFERENCES Vaccine(v_id),
FOREIGN KEY(vis_id) REFERENCES Vision(vis_id),
FOREIGN KEY(s_id) REFERENCES Soldier(s_id)
);

Table created.

SQL> insert into medical values(1,1234,'A+ve',165,68,0,0,1,1);

1 row created.

SQL> select * from medical;

      M_ID       S_ID BG       HEIGHT     WEIGHT    HEARING ISDIABETIC
---------- ---------- ---- ---------- ---------- ---------- ----------
      V_ID     VIS_ID
---------- ----------
         1       1234 A+ve        165         68          0          0
         1          1
```

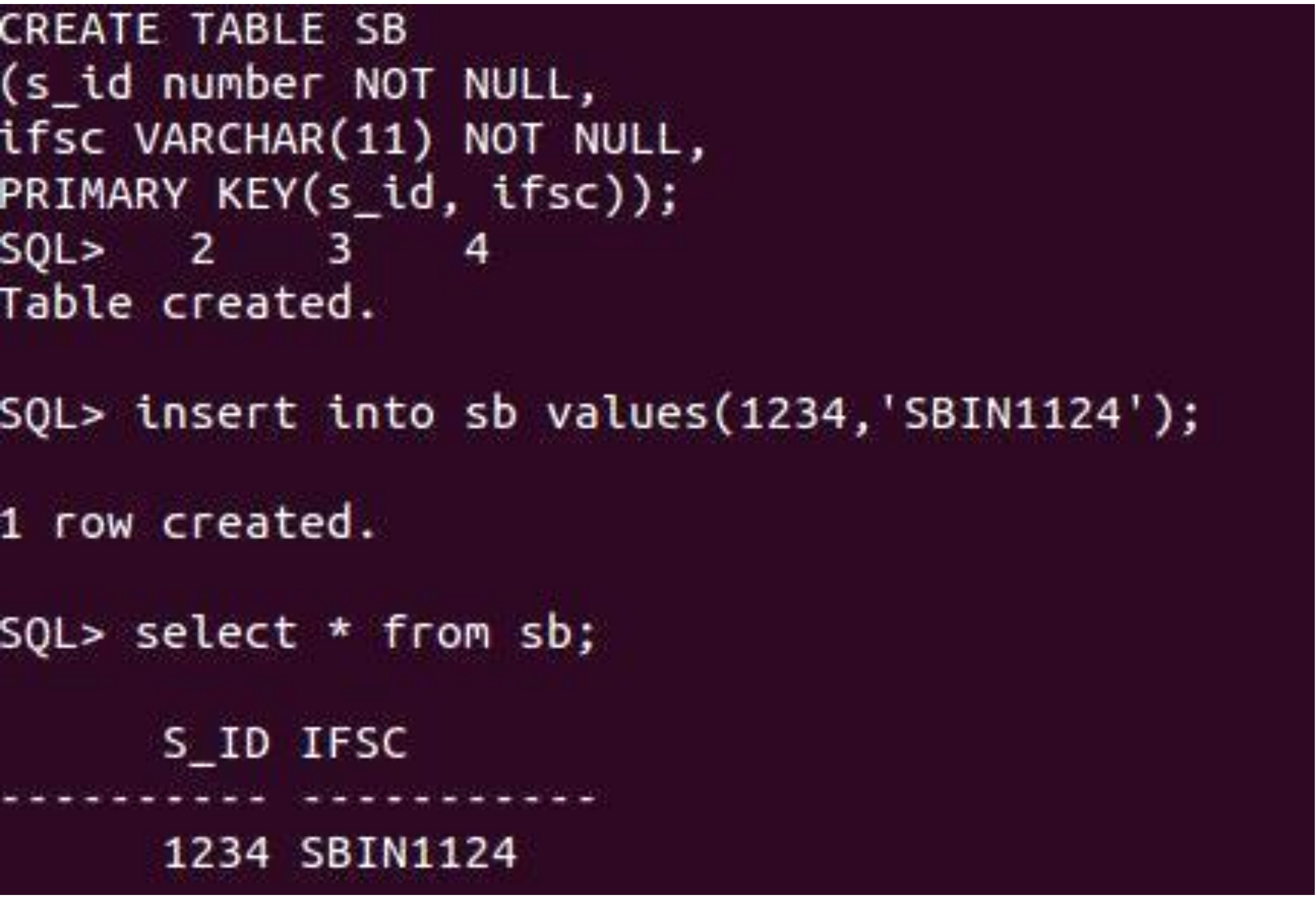


```
SQL> CREATE TABLE Skillset
(
sk_id number NOT NULL PRIMARY KEY,
s_id number NOT NULL,
sprint_speed varchar(20) NOT NULL,
climb_speed varchar(20) NOT NULL,
FOREIGN KEY(s_id) REFERENCES Soldier(s_id)
);  2    3    4    5    6    7    8

Table created.

SQL> insert into skillset values(1,1234,'100 mtrs 13 seconds','50ft 20 secs');

1 row created.

SQL> select * from skillset;

     SK_ID       S_ID SPRINT_SPEED         CLIMB_SPEED
---------- ---------- -------------------- --------------------
         1       1234 100 mtrs 13 seconds  50ft 20 secs
```



```
SQL> CREATE TABLE Bank
(
ifsc VARCHAR(11) NOT NULL PRIMARY KEY,
b_name  VARCHAR(30) NOT NULL,
b_branch VARCHAR(20) NOT NULL
);  2    3    4    5    6

Table created.

SQL> insert into bank values('SBIN1124','State Bank of India','Saharanpur');

1 row created.

SQL> select * from bank;

IFSC        B_NAME                         B_BRANCH
----------- ------------------------------ --------------------
SBIN1124    State Bank of India            Saharanpur
```


```
CREATE TABLE Arms_used
(
a_id number NOT NULL PRIMARY KEY,
s_id number NOT NULL,
AK103 number NOT NULL check(AK103 in(1,0)),
MG56 number NOT NULL check(MG56 in (1,0)),
INAS number NOT NULL check(INAS in(1,0)),
AKM number NOT NULL check(AKM in (1,0)),
Dragunov_svd59 number NOT NULL check(Dragunov_svd59 in (1,0)),
FOREIGN KEY(s_id) REFERENCES Soldier(s_id)
);SQL>   2    3    4    5    6    7    8    9   10   11

Table created.

SQL> insert into arms_used values(1,1234,1,0,0,0,0);

1 row created.

SQL> select * from arms_used;

      A_ID       S_ID      AK103       MG56       INAS        AKM DRAGUNOV_SVD59
---------- ---------- ---------- ---------- ---------- ---------- --------------
         1       1234          1          0          0          0              0
```



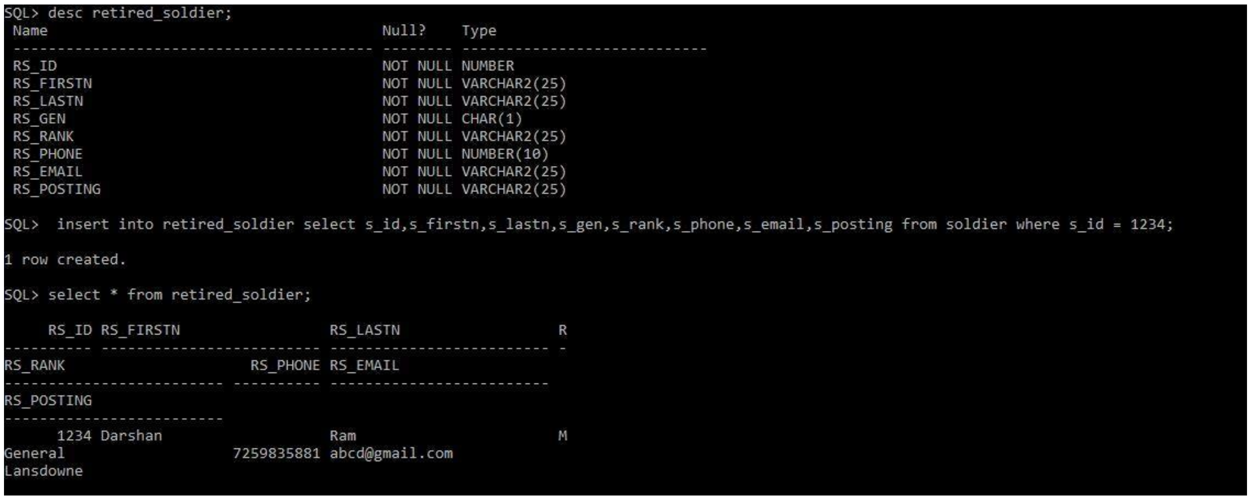
```
CREATE TABLE SB
(s_id number NOT NULL,
ifsc VARCHAR(11) NOT NULL,
PRIMARY KEY(s_id, ifsc));
SQL>   2    3    4
Table created.

SQL> insert into sb values(1234,'SBIN1124');

1 row created.

SQL> select * from sb;

      S_ID IFSC
---------- -----------
      1234 SBIN1124
```

REVIEW 3:

<u>Queries:</u>

1: If the soldier is retired, the maximum percentage of data regarding the present usage and stats will be removed and only the basic data of the military achievements, rank and identity is kept for future references

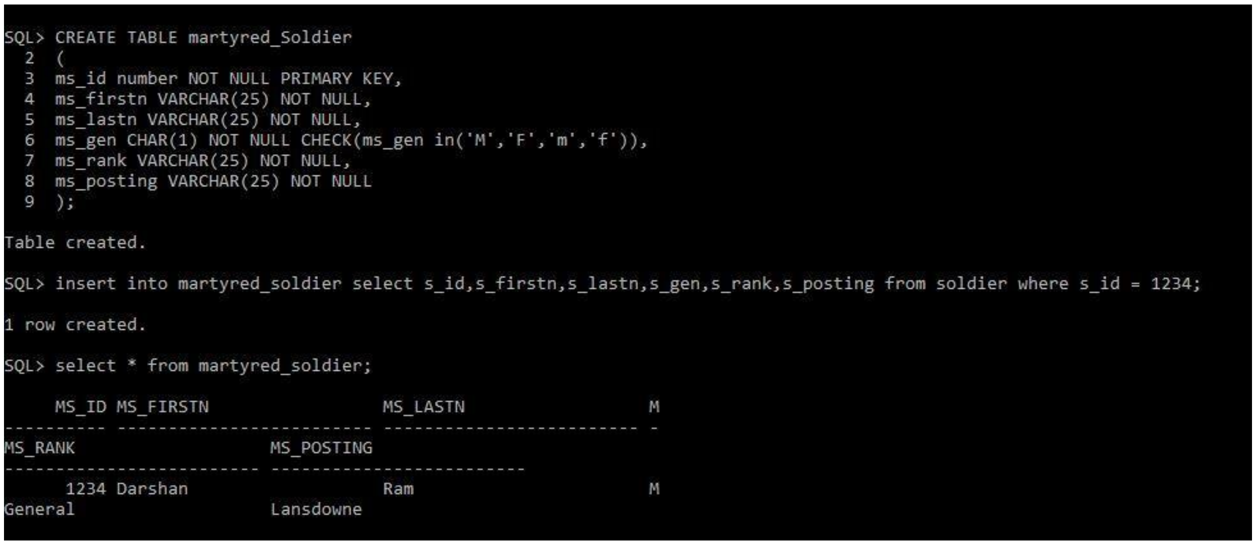
```
CREATE TABLE retired_Soldier
(
rs_id number NOT NULL PRIMARY KEY,
rs_firstn VARCHAR(25) NOT NULL,
rs_lastn VARCHAR(25) NOT NULL,
rs_gen CHAR(1) NOT NULL CHECK(rs_gen in('M','F','m','f')),
rs_rank VARCHAR(25) NOT NULL,
rs_phone number(10) NOT NULL,
rs_email VARCHAR(25) NOT NULL,
rs_posting VARCHAR(25) NOT NULL
);
insert into retired_soldier select s_id,s_firstn,s_lastn,s_gen,s_rank,s_phone,s_email,s_posting
from soldier where s_id = 1234;
delete from soldier where s_id=(select s_id from soldier where s_email = 'abcd@gmail.com');
delete from father where f_id=(select f_id from father where f_email = 'rahuld@gmail.com');
delete from mother where m_id=(select m_id from mother where m_email =
'lmang@gmail.com');
delete from spouse where sp_id=(select sp_id from mother where sp_email =
'rani@gmail.com');
delete from vision where vis_id =1;
delete from vaccine where v_id =1;
delete from medical where m_id =(select s_id from soldier where s_email = 'abcd@gmail.com');
delete from skillset where sk_id =(select s_id from soldier where s_email = 'abcd@gmail.com');
delete from arms_used where a_id =(select s_id from soldier where s_email =
'abcd@gmail.com');
delete from SB where s_id =(select s_id from soldier where s_email = 'abcd@gmail.com');
```



```
SQL> desc retired_soldier;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 RS_ID                                     NOT NULL NUMBER
 RS_FIRSTN                                 NOT NULL VARCHAR2(25)
 RS_LASTN                                  NOT NULL VARCHAR2(25)
 RS_GEN                                    NOT NULL CHAR(1)
 RS_RANK                                   NOT NULL VARCHAR2(25)
 RS_PHONE                                  NOT NULL NUMBER(10)
 RS_EMAIL                                  NOT NULL VARCHAR2(25)
 RS_POSTING                                NOT NULL VARCHAR2(25)

SQL>  insert into retired_soldier select s_id,s_firstn,s_lastn,s_gen,s_rank,s_phone,s_email,s_posting from soldier where s_id = 1234;

1 row created.

SQL> select * from retired_soldier;

     RS_ID RS_FIRSTN                 RS_LASTN                  R
---------- ------------------------- ------------------------- -
RS_RANK                      RS_PHONE RS_EMAIL
------------------------- ---------- -------------------------
RS_POSTING
-------------------------
      1234 Darshan                   Ram                       M
General                   7259835881 abcd@gmail.com
Lansdowne
```

2: If the soldier is martyred in any war or rebellion, the official records essential for present military usage is periodically removed and the past duty records, rank and achievements are stored in the database respectively.

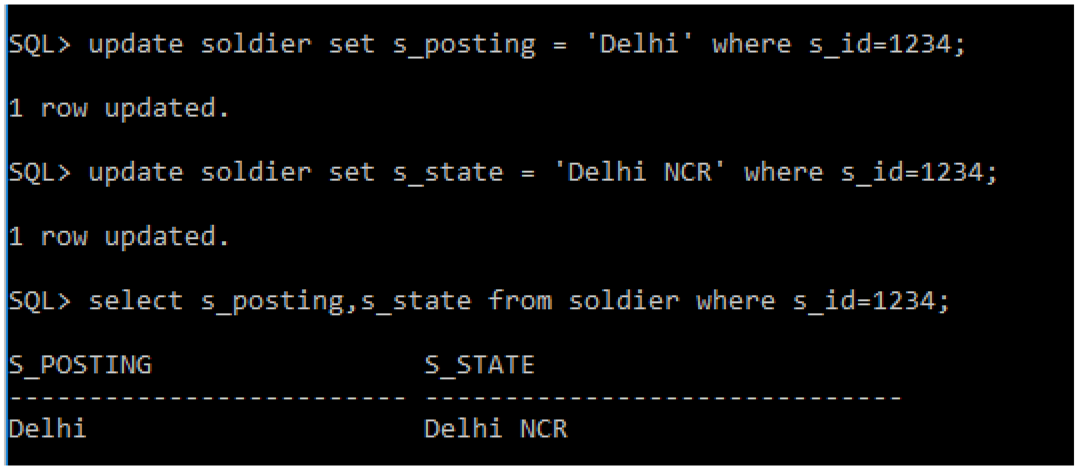
```
CREATE TABLE martyred_Soldier
(
ms_id number NOT NULL PRIMARY KEY,
ms_firstn VARCHAR(25) NOT NULL,
ms_lastn VARCHAR(25) NOT NULL,
ms_gen CHAR(1) NOT NULL CHECK(ms_gen in('M','F','m','f')),
ms_rank VARCHAR(25) NOT NULL,
ms_posting VARCHAR(25) NOT NULL
);
insert into martyred_soldier select s_id,s_firstn,s_lastn,s_gen,s_rank,s_posting from soldier
where s_id = 1234;
delete from soldier where s_id=(select s_id from soldier where s_email = 'abcd@gmail.com');
delete from father where f_id=(select f_id from father where f_email = 'rahuld@gmail.com');
delete from mother where m_id=(select m_id from mother where m_email =
'lmang@gmail.com');
delete from spouse where sp_id=(select sp_id from mother where sp_email =
'rani@gmail.com');
delete from vision where vis_id =1;
delete from vaccine where v_id =1;
delete from medical where m_id =(select s_id from soldier where s_email = 'abcd@gmail.com');
delete from skillset where sk_id =(select s_id from soldier where s_email = 'abcd@gmail.com');
delete from arms_used where a_id =(select s_id from soldier where s_email =
'abcd@gmail.com');
delete from SB where s_id =(select s_id from soldier where s_email = 'abcd@gmail.com');
```



```
SQL> CREATE TABLE martyred_Soldier
  2  (
  3  ms_id number NOT NULL PRIMARY KEY,
  4  ms_firstn VARCHAR(25) NOT NULL,
  5  ms_lastn VARCHAR(25) NOT NULL,
  6  ms_gen CHAR(1) NOT NULL CHECK(ms_gen in('M','F','m','f')),
  7  ms_rank VARCHAR(25) NOT NULL,
  8  ms_posting VARCHAR(25) NOT NULL
  9  );

Table created.

SQL> insert into martyred_soldier select s_id,s_firstn,s_lastn,s_gen,s_rank,s_posting from soldier where s_id = 1234;

1 row created.

SQL> select * from martyred_soldier;

     MS_ID MS_FIRSTN                 MS_LASTN                  M
---------- ------------------------- ------------------------- -
MS_RANK                   MS_POSTING
------------------------- -------------------------
      1234 Darshan                   Ram                       M
General                   Lansdowne
```

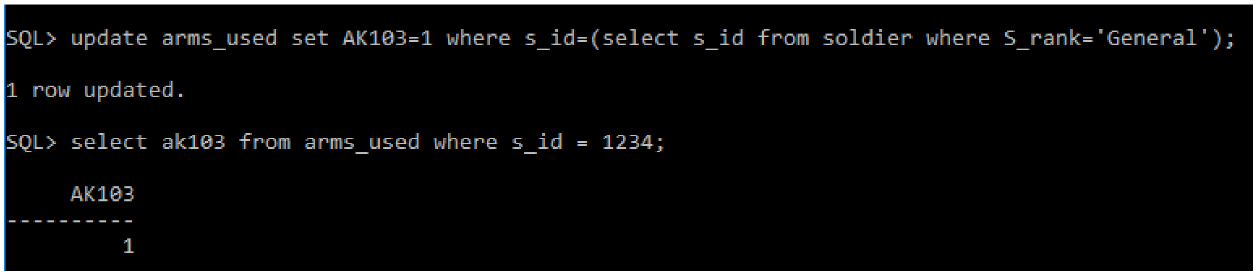
3: If the soldier's field of duty is transferred to a different military regiment or posted or stationed to different battalion, the data is modified accordingly.

update soldier set s_posting = 'Delhi' where s_id=1234;
update soldier set s_state = 'Delhi NCR' where s_id=1234;



```
SQL> update soldier set s_posting = 'Delhi' where s_id=1234;

1 row updated.

SQL> update soldier set s_state = 'Delhi NCR' where s_id=1234;

1 row updated.

SQL> select s_posting,s_state from soldier where s_id=1234;

S_POSTING                 S_STATE
------------------------- ------------------------------
Delhi                     Delhi NCR
```

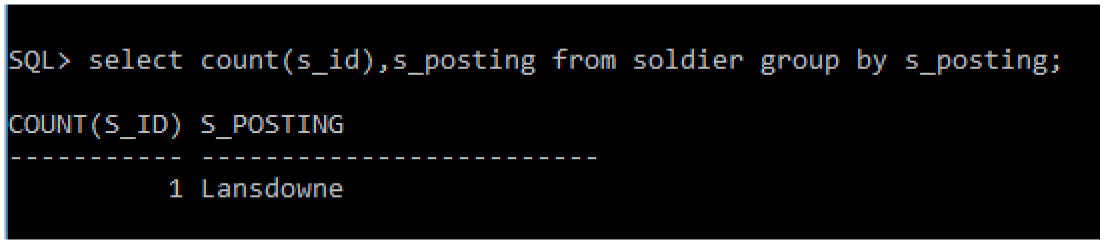
4: The soldier's weapon should be appropriate to his rank.
update arms_used set AK103=1 where s_id=(select s_id from soldier where S_rank='General');



```
SQL> update arms_used set AK103=1 where s_id=(select s_id from soldier where S_rank='General');

1 row updated.

SQL> select ak103 from arms_used where s_id = 1234;

     AK103
----------
         1
```

5:The number of soldiers posted at a particular posting area. For example, Lansdowne.

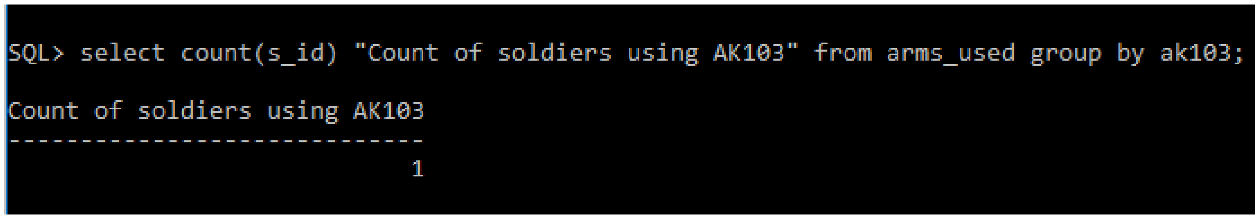
select count(s_id),s_posting from soldier group by s_posting;



```
SQL> select count(s_id),s_posting from soldier group by s_posting;

COUNT(S_ID) S_POSTING
----------- -------------------------
          1 Lansdowne
```

6: To find the number of soldiers carrying a particular weapon.

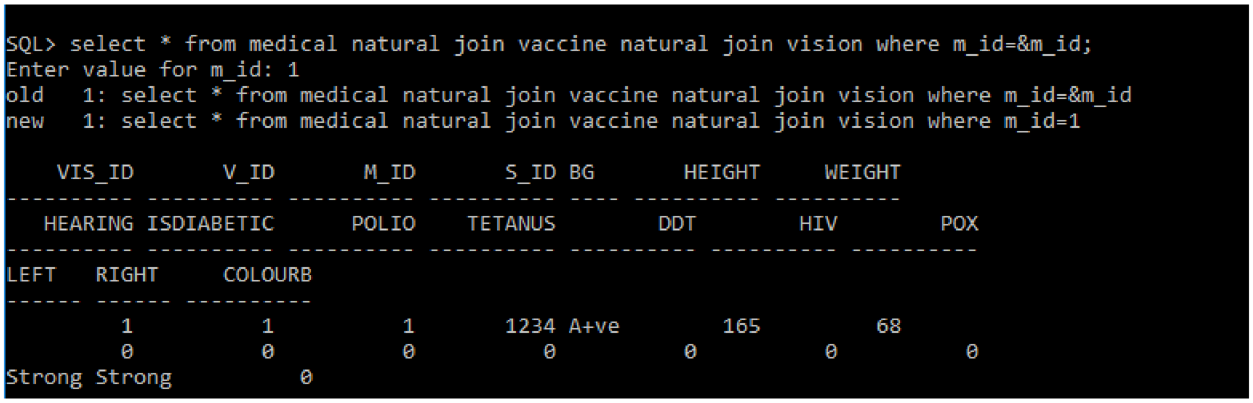
select count(s_id) "Count of soldiers using AK103" from arms_used group by ak103;



```
SQL> select count(s_id) "Count of soldiers using AK103" from arms_used group by ak103;

Count of soldiers using AK103
-----------------------------
                            1
```

7: Regular check-ups of the soldier have to be done and compared with the previous results of the soldier for which the medical history has to be retrieved from the database.

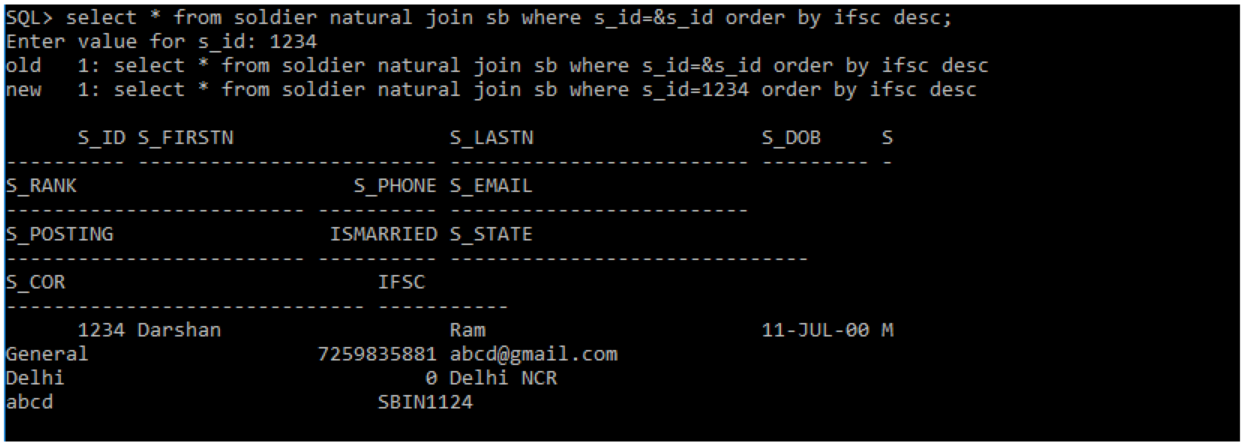
select * from medical natural join vaccine natural join vision where m_id=&m_id;



```
SQL> select * from medical natural join vaccine natural join vision where m_id=&m_id;
Enter value for m_id: 1
old   1: select * from medical natural join vaccine natural join vision where m_id=&m_id
new   1: select * from medical natural join vaccine natural join vision where m_id=1

    VIS_ID       V_ID       M_ID       S_ID BG       HEIGHT     WEIGHT
---------- ---------- ---------- ---------- ---- ---------- ----------
   HEARING ISDIABETIC      POLIO    TETANUS        DDT        HIV        POX
---------- ---------- ---------- ---------- ---------- ---------- ----------
LEFT   RIGHT     COLOURB
------ ------ ----------
         1          1          1       1234 A+ve        165         68
         0          0          0          0          0          0          0
Strong Strong          0
```

8: When soldiers are paid, the bank details of the soldiers must be retrieved with the soldier's details, in descending order of the IFSC code.

select * from soldier natural join sb where s_id=&s_id order by ifsc desc;

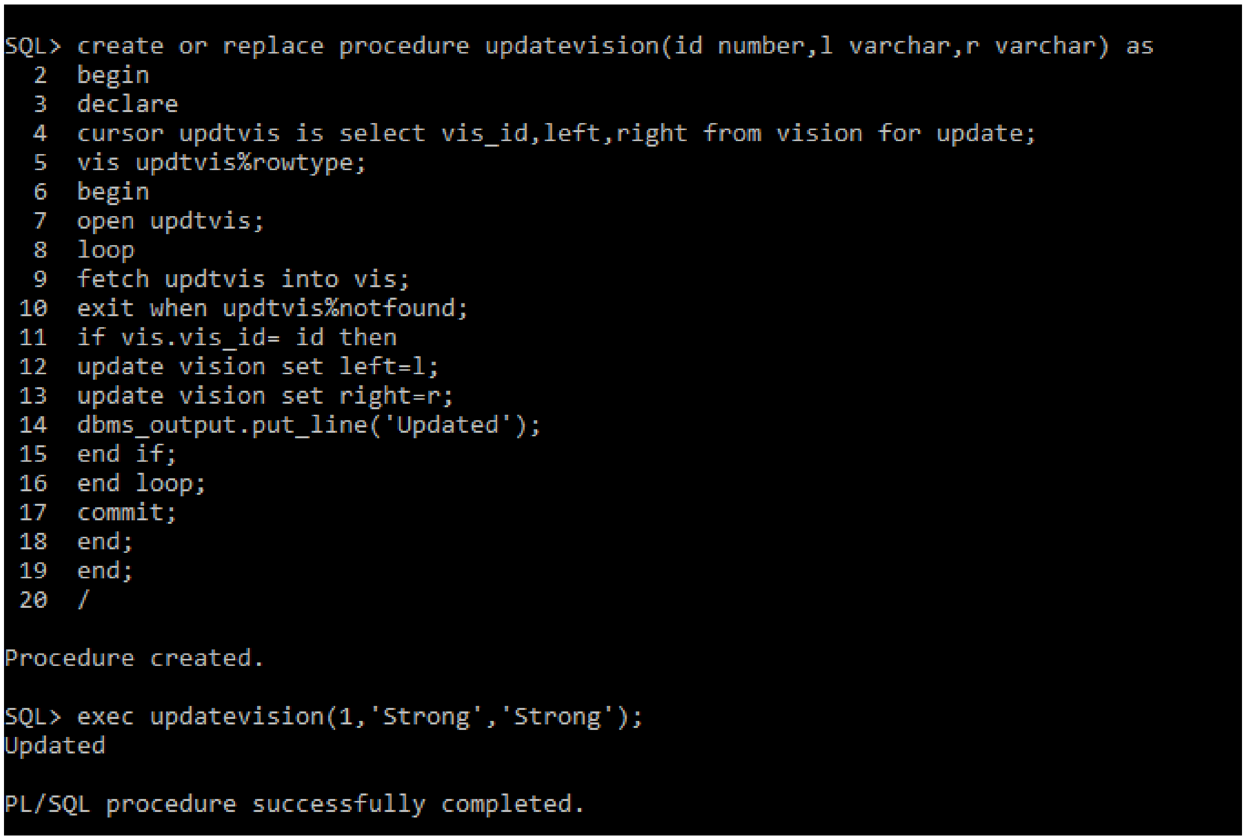


```
SQL> select * from soldier natural join sb where s_id=&s_id order by ifsc desc;
Enter value for s_id: 1234
old   1: select * from soldier natural join sb where s_id=&s_id order by ifsc desc
new   1: select * from soldier natural join sb where s_id=1234 order by ifsc desc

      S_ID S_FIRSTN                 S_LASTN                  S_DOB     S
---------- ------------------------ ------------------------ --------- -
S_RANK                      S_PHONE S_EMAIL
------------------------ ---------- ------------------------
S_POSTING                 ISMARRIED S_STATE
------------------------ ---------- -----------------------------
S_COR                          IFSC
------------------------------ -----------
      1234 Darshan                  Ram                      11-JUL-00 M
General                  7259835881 abcd@gmail.com
Delhi                             0 Delhi NCR
abcd                           SBIN1124
```

PL/SQL Procedure and Function:

1: Procedure to update the table Vision.

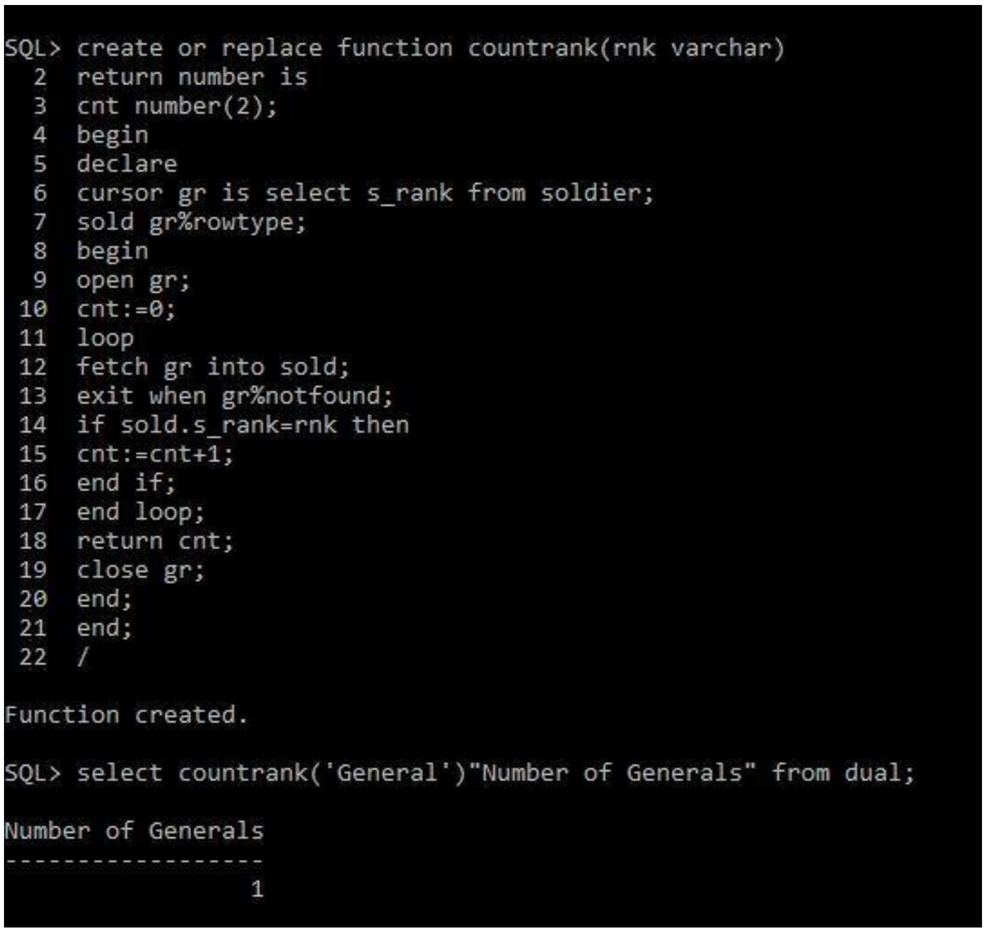
```
create or replace procedure updatevision(id number,l varchar,r varchar) as
begin
declare
cursor updtvis is select vis_id,left,right from vision for update;
vis updtvis%rowtype;
begin
open updtvis;
loop
fetch updtvis into vis;
exit when updtvis%notfound;
if vis.vis_id= id then
update vision set left=l;
update vision set right=r;
dbms_output.put_line('Updated');
end if;
end loop;
commit;
end;
end;
/
```



```
SQL> create or replace procedure updatevision(id number,l varchar,r varchar) as
begin
declare
cursor updtvis is select vis_id,left,right from vision for update;
vis updtvis%rowtype;
begin
open updtvis;
loop
fetch updtvis into vis;
exit when updtvis%notfound;
if vis.vis_id= id then
update vision set left=l;
update vision set right=r;
dbms_output.put_line('Updated');
end if;
end loop;
commit;
end;
end;
/

Procedure created.

SQL> exec updatevision(1,'Strong','Strong');
Updated

PL/SQL procedure successfully completed.
```

2: Function to get the rank of the soldier.

```
create or replace function countrank(rnk varchar)
return number is
cnt number(2);
begin
declare
cursor gr is select s_rank from soldier;
sold gr%rowtype;
begin
open gr;
cnt:=0;
loop
fetch gr into sold;
exit when gr%notfound;
if sold.s_rank=rnk then
cnt:=cnt+1;
end if;
end loop;
return cnt;
close gr;
end;
end;
/
```

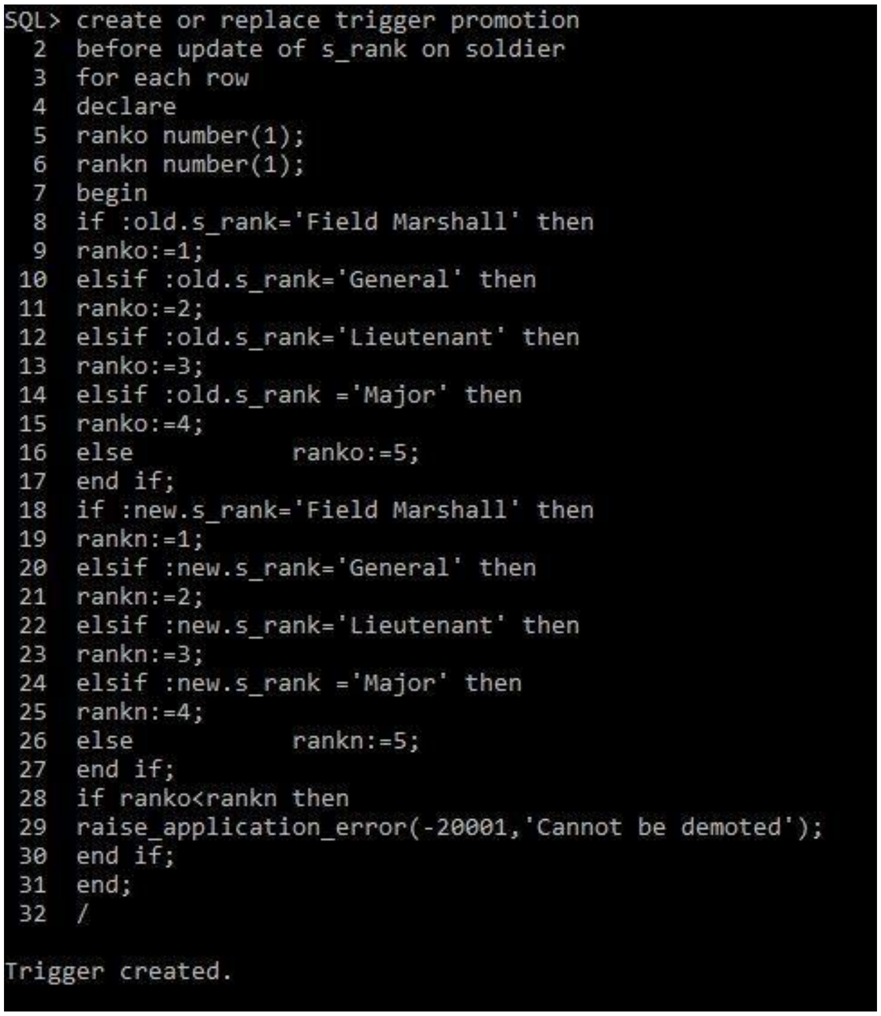


```
SQL> create or replace function countrank(rnk varchar)
  2  return number is
  3  cnt number(2);
  4  begin
  5  declare
  6  cursor gr is select s_rank from soldier;
  7  sold gr%rowtype;
  8  begin
  9  open gr;
 10  cnt:=0;
 11  loop
 12  fetch gr into sold;
 13  exit when gr%notfound;
 14  if sold.s_rank=rnk then
 15  cnt:=cnt+1;
 16  end if;
 17  end loop;
 18  return cnt;
 19  close gr;
 20  end;
 21  end;
 22  /

Function created.

SQL> select countrank('General')"Number of Generals" from dual;

Number of Generals
------------------
                 1
```

Trigger to implement business rules:

1: New rank given cannot be lesser than the old rank.

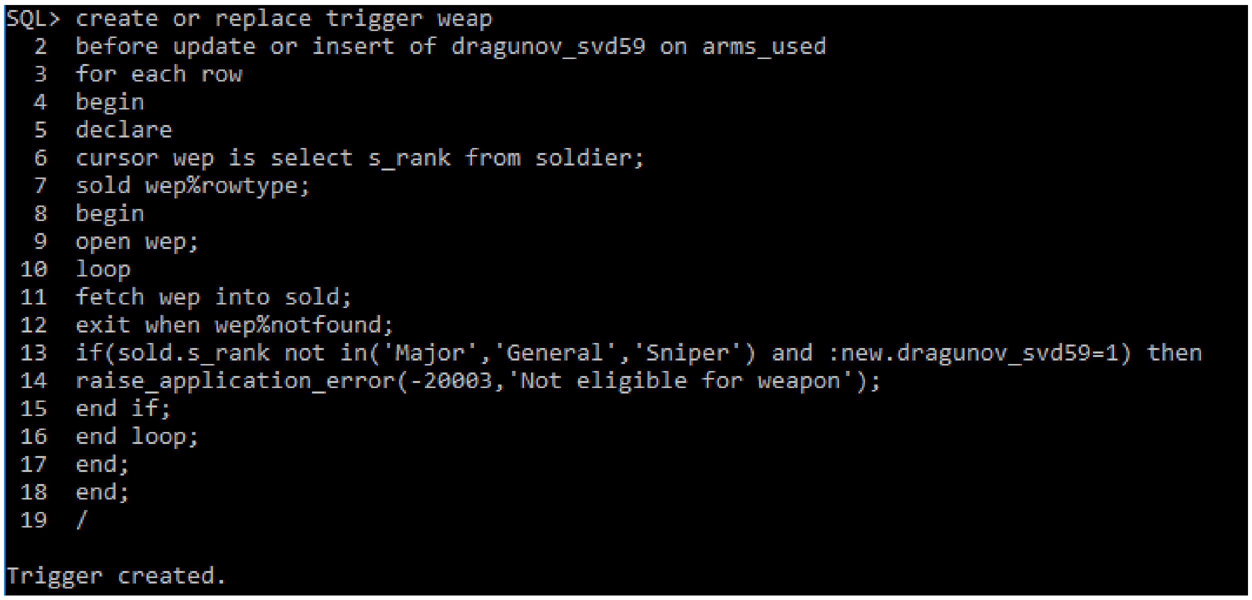
```
create or replace trigger promotion
before update of s_rank on soldier
for each row
declare
ranko number(1);
rankn number(1);
begin
if :old.s_rank='Field Marshall' then
        ranko:=1;
elsif :old.s_rank='General' then
        ranko:=2;
elsif :old.s_rank='Lieutenant' then
        ranko:=3;
elsif :old.s_rank ='Major' then
        ranko:=4;
else        ranko:=5;
end if;
if :new.s_rank='Field Marshall' then
        rankn:=1;
elsif :new.s_rank='General' then
        rankn:=2;
elsif :new.s_rank='Lieutenant' then
        rankn:=3;
elsif :new.s_rank ='Major' then
        rankn:=4;
else        rankn:=5;
end if;
if ranko<rankn then
raise_application_error(-20001,'Cannot be demoted');
end if;
end;
/
```



```
SQL> create or replace trigger promotion
before update of s_rank on soldier
for each row
declare
ranko number(1);
rankn number(1);
begin
if :old.s_rank='Field Marshall' then
ranko:=1;
elsif :old.s_rank='General' then
ranko:=2;
elsif :old.s_rank='Lieutenant' then
ranko:=3;
elsif :old.s_rank ='Major' then
ranko:=4;
else           ranko:=5;
end if;
if :new.s_rank='Field Marshall' then
rankn:=1;
elsif :new.s_rank='General' then
rankn:=2;
elsif :new.s_rank='Lieutenant' then
rankn:=3;
elsif :new.s_rank ='Major' then
rankn:=4;
else           rankn:=5;
end if;
if ranko<rankn then
raise_application_error(-20001,'Cannot be demoted');
end if;
end;
/

Trigger created.
```

2: Soldier cannot be assigned a sniper if he doesn’t fall under the ranks of Major, General or Sniper

```
create or replace trigger weap
before update or insert of dragunov_svd59 on arms_used
for each row
begin
declare
cursor wep is select s_rank from soldier;
sold wep%rowtype;
begin
open wep;
loop
fetch wep into sold;
exit when wep%notfound;
if(sold.s_rank not in('Major','General','Sniper') and :new.dragunov_svd59=1) then
raise_application_error(-20003,'Not eligible for weapon');
end if;
end loop;
end;
end;
/
```



```
SQL> create or replace trigger weap
  2  before update or insert of dragunov_svd59 on arms_used
  3  for each row
  4  begin
  5  declare
  6  cursor wep is select s_rank from soldier;
  7  sold wep%rowtype;
  8  begin
  9  open wep;
 10  loop
 11  fetch wep into sold;
 12  exit when wep%notfound;
 13  if(sold.s_rank not in('Major','General','Sniper') and :new.dragunov_svd59=1) then
 14  raise_application_error(-20003,'Not eligible for weapon');
 15  end if;
 16  end loop;
 17  end;
 18  end;
 19  /

Trigger created.
```