

## Assignment #6

1. CREATE TABLE EMPLOYEE

```
(Fname      VARCHAR(15)      NOT NULL,  
 Minit      CHAR,  
 Lname      VARCHAR(15)      NOT NULL,  
 Ssn        CHAR(9)          NOT NULL,  
 Bdate      DATE,  
 Address    VARCHAR(40),  
 Sex        CHAR,  
 Salary     DECIMAL(10,2),  
 Super_ssn  CHAR(9),  
 Dno        INT              NOT NULL,  
 PRIMARY KEY (Ssn),  
 FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn));
```

2. In order to create the view I had to grant my user an additional permission. The permission was CREATE VIEW.

3. Question 5.9

a. This statement would be allowed.

Statement for base table:

```
SELECT Dno, COUNT(*), SUM(Salary), AVG(Salary)  
FROM EMPLOYEE  
GROUP BY Dno;
```

Results from figure 3.6:

5	4	133000	33250
4	3	93000	31000
1	1	55000	55000

b. This statement would be allowed.

Statement for base table:

```
SELECT Dno, COUNT(*)  
FROM EMPLOYEE  
GROUP BY Dno  
HAVING SUM(Salary) > 100000;
```

Results from figure 3.6:

5	4
---	---

c. This statement would be allowed.

Statement for base table:

```
SELECT Dno, AVG(Salary)  
FROM EMPLOYEE  
GROUP BY Dno
```

## Assignment #6

```
HAVING COUNT(*) > (SELECT COUNT(*)  
FROM EMPLOYEE  
WHERE Dno=4);
```

Results from figure 3.6:

5	33250
---	-------

- d. This statement would not be allowed because its attempting to update a view which contains aggregate functions and a GROUP BY clause.

Statement for base table:

```
UPDATE EMPLOYEE  
SET Dno=3  
WHERE Dno=4;
```

Results from figure 3.6:

Alicia, Jennifer, and Ahmad would have their Dno changed.

- e. This statement would not be allowed because its attempting to delete from a view which contains aggregate functions and a GROUP BY clause.

Statement for base table:

```
DELETE FROM EMPLOYEE  
WHERE Dno  
IN (SELECT Dno  
FROM EMPLOYEE  
GROUP BY Dno  
HAVING COUNT(*) > 4);
```

Results from figure 3.6:

John, Franklin, Ramesh, and Joyce would have their rows deleted.

4. This lab was really interesting. It's amazing to see how creating a view greatly simplifies the SQL statements needed generate queries. While generating the fake data for my database I ended up finding a very neat python library called fake-factory. This library helped me generate random unique data I could use in my database so with a little work in python I was able to create a thousand rows of fake data in a couple minutes.

Screenshots of my database query results:

## Assignment #6

```
Terminal - ec2-user@ip-172-31-25-45:~
File Edit View Terminal Tabs Help

SQL> SELECT * FROM DEPT_SUMMARY;
-----
D      C      TOTAL_S  AVERAGE_S
-----
1      192  53613346.3  501059.311
2      207  51958414.8  514439.75
5      197  51843271.9  525012.978
4      193  47848632.1  480251.756
3      195  49009730.4  500097.29

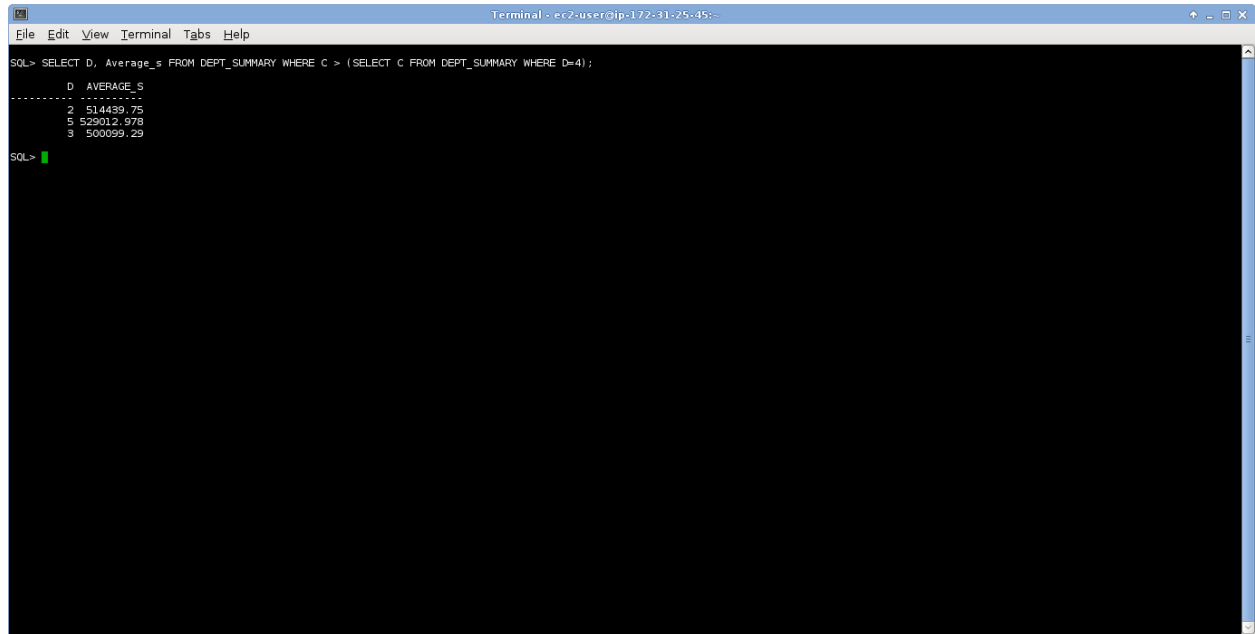
SQL>
```

```
Terminal - ec2-user@ip-172-31-25-45:~
File Edit View Terminal Tabs Help

SQL> SELECT D, C FROM DEPT_SUMMARY WHERE Total_s > 100000;
-----
D      C
-----
1      192
2      207
5      197
4      193
3      195

SQL>
```

## Assignment #6



The image shows a terminal window with a blue title bar and a menu bar. The title bar text is "Terminal - ec2-user@ip-172-31-25-45:~". The menu bar contains "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal content shows an SQL query being executed, followed by a header line "D AVERAGE\_S" and three rows of data. The prompt "SQL>" is visible at the bottom left.

```
SQL> SELECT D, Average_s FROM DEPT_SUMMARY WHERE C > (SELECT C FROM DEPT_SUMMARY WHERE D=4);  
D AVERAGE_S  
-----  
2 514439.75  
5 529012.978  
3 500099.29  
SQL>
```

## Assignment #6

```
Terminal - ec2-user@ip-172-31-25-45:~
File Edit View Terminal Tabs Help

SQL> UPDATE DEPT_SUMMARY SET D=3 WHERE D=4;
UPDATE DEPT_SUMMARY SET D=3 WHERE D=4
      *
ERROR at line 1:
ORA-01732: data manipulation operation not legal on this view

SQL>
```

```
Terminal - ec2-user@ip-172-31-25-45:~
File Edit View Terminal Tabs Help

SQL> DELETE FROM DEPT_SUMMARY WHERE C > 4;
DELETE FROM DEPT_SUMMARY WHERE C > 4
      *
ERROR at line 1:
ORA-01732: data manipulation operation not legal on this view

SQL>
```