**Q&A SQL Fundamentals Part III**

1.) Create at table called emp\_test. Get the structure and the data from hr.employees

table.

2.) Create a table called emp\_test2. The specification of the table is:

• Firstname varchar2(20)

• Lastname varchar2(20)

• employee\_id number

• userid varchar2(20)

• salary number

3.) Create a product\_master tabl

• Price number (5,2)

4.) Create a table called emp2 in the scott schema. These are the specifications.

• SSN number (9) Primary Key (Give it a meaningful constraint name.)

• First\_name Varchar2(25) with a not null constraint

• Last\_name Varchar2(25) with a not null constraint

• Emp\_id number (6) Unique key (Give it a meaningful constraint

name.)

• Salary number (9,2)

• Hire\_date Date with a not null constraint.

5.) Add a PRIMARY KEY constraint to the EMP\_TEST2 table you created earlier on

the employee\_id column. The constraint should be named at creation. Name the

constraint emp\_id\_pk.

6.) Add the first row of data to the Emp\_test2 table from the following sample data.

Do not list the columns in the INSERT clause. Make it permanent.

Emp ID LAST\_NAME FIRST\_NAME USERID SALARY

1 Johnson Charles cjohnson 771

2 Smith Tim tsmith 819

3 Lucas Frank Flucas 1195

4 Thomas David DThomas 790

5 Thompson Ashley athompson 1400

6 Bryant Kobe kbryant 1900

7 Jordan Michael mjordan 2100

Property of Neway IT Solutions

7.) Populate the Emp\_test2 table with the second row of sample data from the list

below. This time, list the columns explicitly in the INSERT clause. Make it

permanent.

ID LAST\_NAME FIRST\_NAME USERID SALARY

8 Lolade Johnson ljohnson 558

9 Ebreu Richard rebreu 1024

10 Winfrey Oprah owinfrey 958

8.) In the emp\_test2 table, change the lastname of employee 4 to Tom. Do NOT Make

it permanent.

9) Revert employee 4’s name back to what it was originally.

10) In the emp\_test2 table, change the lastname of employee 7 to Tom. Make it

permanent.

9.) In the emp\_test2 table change the salary to 1000 for all employees with a salary

less than 900 Make it permanent. You decided to give them all a raise.

10.) In the emp\_test2 table, delete the Oprah Winfrey

16.) Permanently get rid of the data in the scott.emp4 table.

17.) Populate the scott.emp4 table. Use scott.emp table as the source of your

data.

18.) Rename the emp4 table to your test5.

19.) List 2 differences between Primary key and Unique key.

20.) Drop the unique key constraint in the emp2 table you created earlier 21.) Disable

the primary key constraint on the product\_master table you

created earlier.

22.) Add a sex column with varchar2(1) data type to the emp2 table you created

earlier.

23.) Increase the first and last name column length to Varchar2(30)each on

the emp\_test2 table you created earlier.

24.) . Create a view called EMPLOYEES\_VW based on the employee numbers,

employee names, and department numbers from the hr.EMPLOYEES table. Change

the heading for the employee name to EMPLOYEE.

25.) Using your EMPLOYEES\_VW view, enter a query to display all

employee names and department numbers and employee\_id

26.) Create a view named DEPTVW that contains the employee numbers, employee

last names, and department numbers for all employees in hr.employees table.

50. Label the view columns EMPNO, EMPLOYEE, and DEPTNO ( hint: use alias

when labelling the column names)

27.) Create a sequence to be used with the primary key column of the DEPT table.

The sequence should start at 300 and have a maximum value of 1000. Have your

sequence increment by ten numbers. Name the sequence DEPT \_SEQ.

28.) Create an index on the last\_name column of the scott.emp2 table you

created earlier. Call the index lname\_idx

Property of Neway IT Solutions

29.) Create a public synonym for the scott.emp table;

30.) Create a composite index on first\_name, last\_name of the employee table.

31.) Drop the product\_master table you created earlier in the scott schema.

32.) Create the table called FACULTY with the following colums ( FID,

I\_NAME, I\_LOC ).Make FID colum the primary key column

33.) Create the table called SPECIALTY with the following colums( SPID,

SP\_DESC ) Make SPID column the primary key column

34.) Create the table called FAC\_SPECIALTY with the follwing colums (

FID, SPID ) Make FID and SPID the foreign keyS of the Faculty and

Specialty tables

35.) Create the table called STUDENT with the following colums ( SID,

NAME, MAJOR, GRADE\_LEVEL, AGE, ADVISOR ) Make SID the primary key.

Also make the ADVISOR column the foreign key to the FID column in the Faculty

table.

36.) Create the table called CLASS with the following colums (C\_NAME,

TIME, ROOM, FID)Make the C\_NAME column the primary key. Also make the FID

column the foreign key to the FID column in the Faculty table.

37.) Create the table called ENROLLMENT with the following colums (

SID, C\_NAME ) Make the C\_NAME column a foreign key to reference C\_NAME on

Class table. Also make the SID column to reference the SID column on the Student

table.

38.) The FACULTY table should contain the following data

FID I\_NAME I\_LOC

100 AKIN 100BC

Property of Neway IT Solutions

200 SMITH 200BC

300 WILLIAMS 300BC

400 RONDER 400AD

500 ROUSEY 500AD

600 TYSON 600BC

700 SMITH

800 KENNY

900 BALIS 900AD

39.) The specialty table should contain the following data :

SPID SP\_DESC

001 Firewall

002 Database Admin

003 End User Migration

004 Defense Support

005 Expert System Maintenance

006 Information Technology

007 Data Base Management Systems

008 Quality Assurance

009 Visual Basic

010 Database Development Support

011 Defense Management

012 Marine Reconnaisance

Property of Neway IT Solutions

40.) The FAC\_SPECIALTY table should contain the

200 THOMAS MIS FR 21 100

250 SAM MKT S R 24 200

300 CLARA MKT SR 28 300

350 MORRIS MIS GR 28

400 RIDER MKT SR 30 100

450 JONAS SR

500 CARTER MGMT 100

700 AMBER

800 SMITH ACCT FR 18 200

900 THOMAS BIO SR 25 100

42.) The CLASS table should contain the following data

C\_NAME TIME ROOM FID

INSS222 7:30M 201BC 300

INSS301 9:35M 220BC 200

INSS225 7:30M 220BC

INSS421 6:45M 330BC 400

INSS499 2:30P 100

MKT461 8:30W 331BC 600

MKT499 10.45M 500AD 400

INSS620 5:30T 220BC 300

INSS641 9:50TH 333BC 100

INSS111 7:15W 355BC 200

Property of Neway IT Solutions

43.) The ENROLLMENT table should contain the following data

SID C\_NAME

100 INSS222

200 INSS111

100 INSS651

350 INSS610

350 INSS651

200 INSS301

200 INSS421

400 MKT620

400 INSS499

400 MKT421

450 INSS225

450 INSS301

100 INSS641

700 INSS111

700 INSS499

800 MKT461

900 INSS301

From the above 6 tables

44.) Return a list of all the student names and the class name(s) they are enrolled in.

45.) Return a list of all the student names, major, and their advisor’s names.

Property of Neway IT Solutions

46.) Return a list of all faculty names and the specialties each faculty has.

47.) Return a list of faculty IDs and faculty names that have a specialty of “MARINE

RECONNAISANCE”.

48.) Return a list of student names and student age that are taking classes taught by

faculty member “TYSON”

49.) Return a list of all the student names and all the faculty names.

50.) Duplicate the following tables: Student, Class and Faculty. Call

the names Student2, Class2 and Faculty2. Make sure that the duplicate tables have the

same data in the original table.

51.) Drop the 3 duplicate tables you just created.

52.) Create a table called test8. Let the data and structure come from

the scott.emp table.

53.) SMITH Just got promoted. His new salary is now 3200, his new

position is salesman and his new department is 40. Issue a single update state to effect

the new changes in the test8 table.

54.) Increase the salary and job title of James to 72000 and BOSS

in the test8 table. Make sure you save it.

55.) In the test8 table. Use a single query to get rid of all the

employee data in department 20 except for the President. Make sure you save the data.

56.) In the test8 table, get rid of all the employees that make commission. Do not save

it.

57.) Issue a command to get back all the employees that make commission.

58.) Get rid of all the data in the test8 table. Make sure you save it.

59.) Issue a command to rename the test8 table to test9.

Property of Neway IT Solutions

60.) Get rid of the test9 table

61.) What is a primary key?

62.) What is a foreign key?

63) What is the difference between a primary key and a unique key?

64) What is the relationship between a primary and foreign key?

65) What is the purpose of the check constraint?

66) Using a join statement list the employee lastname, firstname, departmentname,

locationid in one join statement

67) Show me all the employees that earn more than Chung in one statement ( hint: we

are NOT using join here)

68) Show me all the employees that are in the same department as Whalen in the

employees tables

69) Show me the ename and department names from the emp and dept tables in the

scott schema

70) Show me the employee numbers, job, loc using the emp, dept tables in the scott

schema. (Hint: to find all the tables that a user has describe the : dba\_tables data

dictionary)

71) Show me all the data in the salgrade table owned by the scott schema.

72) Show me the orederid, productid in the order\_items table in the oe schema.

73) Show me the unitprice, autity, order date, order mode from the orders and

order\_items tables in the oe schema.

74) Show me the line itme id, cutomer\_id, warehouse\_id from the orders, order items

and inventory tables of the oe schema.

75) show me the salesrepid, , marital status, cust\_first\_name,cust\_last\_name from the

customers and orders table

76) Show me the warehousespec, warehousename, quatity on hand from the

warehouse and inventory tables in the oe schema

Property of Neway IT Solutions

77) Show me all the data in the product\_description table owned by oe schmas

78) Show me all the data in the product\_information table owned by the oe schema

79) Show me all the translated\_name, weight\_class, supplier\_id, product\_status ,

listprice, lin item id from the product information table, product description table and

the order\_items tables.

80) Show me all the data in the orders tables

81) Create a view called orders\_vwx as the order\_id, orderdate and promotionid of the

orders table owned by the oe schema. Ensure that the view is owned by hr.

82) create a view called orders\_vwx2 as order\_total, linitemis, quatity on hand . Query

the orers, inventory and order\_items tables owned by oe. Ensure that the view is

owned by scott.

83) create a view called warehouse\_vw as the the warehouse\_name, product\_id from

the warehouse and inventory tables in the oe schema. Ensure that sys owns the view.

84) show me product description, translated\_name, translated decription, weight class,

warehouse id from inventories, product description, product information tables owned

by the oe schema. (Hint: use a join of 3 tables)

85) Drop the orders\_vwx view

86) Drop the orders\_vwx2 view

87) Delete all the datain the warehouse\_vw view. Do NOT make it permanent.

88) Get the data back.

89) drop view warehouse\_vw

90) create a new table called warehouse\_new as translated\_name, weight\_class,

supplier\_id, product\_status , listprice, lin item id from the product information table,

product description table and the order\_items tables.

91) Show me all the data in the warehouse\_new table

92) create a new table called orders\_new as salesrepid, , marital status,

cust\_first\_name,cust\_last\_name. Ensure that the table is owned by the hr schemas

Property of Neway IT Solutions

93) Show me all the data in the order\_new table .

94) delete all the data in the order\_new table but do not make it permanent.

95) retrieve all the data back

96) remove all the data in the order\_new table in such a way that the structure is intact

but the data is gone forever

97) Drop the structure and table of the order\_new table permanently

98) Query the order\_new table to ensure that you cannot see any data or structure. Let

me see it

99) Create a new table called employees\_new showing me the last\_name,

deparment\_name, city, phonenumber, postalcode from the employees, departments

and locations tables. Ensure the table is owned by the oe schema.

100) Permanently get rid of the employees\_new table.

Please send your project answers to newayitsolutions@gmail.com

P

1. SQL> create table emp\_test as select \* from hr.employees;

2. SQL> create table emp\_test2 (Firstname varchar2(20), Lastname varchar2(20), employee\_id number, userid varchar2(20), salary number);

3. SQL> create table scott.product\_master (Product\_id number(7), Product\_name varchar2(50) not null, Code varchar2(10) not null, Reorder\_threshold number(5), Cost number (5,2), Price number (5,2), constraint PK\_INV primary key(Product\_id), constraint chk\_reorder check(Reorder\_threshold > 0));

4. SQL> create table scott.emp2 (SSN number (9), First\_name Varchar2(25) not null, Last\_name Varchar2(25) not null, Emp\_id number (6), Salary number (9,2), Hire\_date Date not null, constraint PK\_SSN primary key(SSN), constraint UK\_empid unique(Emp\_id));

5. SQL> alter table emp\_test2 add constraint emp\_id\_pk primary key(employee\_id);

6. SQL> insert into emp\_test2 values ('Ashley', 'Thompson', 5, 'athompson', 1400);

1 row created.

SQL> commit;

Commit complete.

7. SQL> insert into emp\_test2 (FIRSTNAME, LASTNAME, EMPLOYEE\_ID, USERID, SALARY) values ('Johnson', 'Lolade', 8, 'ljohnson', 558);

1 row created.

SQL> commit;

8. SQL> update emp\_test2 set LASTNAME = 'Tom' where EMPLOYEE\_ID = 4;

9. SQL> rollback;

10. SQL> update emp\_test2 set LASTNAME = 'Tom' where EMPLOYEE\_ID = 7;

1 row updated.

SQL> commit;

11. SQL> update emp\_test2 set SALARY = 1000 where SALARY < 900;

4 rows updated.

SQL> commit;

12. SQL> delete from emp\_test2;

13. SQL> rollback;

14. When you drop a table, it clears the structure and data of the table. When you delete or truncate, it only clears the data; the structure of the table still remains. "Delete" is DML, "Drop" and "Truncate" are DDL. You can use rollback after a delete, but you can't rollback after performing drop or truncate.

15. SQL> create table scott.emp3 as select \* from scott.emp;

16. SQL> create table scott.emp4 as select empno, ename, job, sal, deptno from scott.emp;

17. SQL> delete from scott.emp4;

14 rows deleted.

SQL> commit;

Commit complete.

18. SQL> insert into scott.emp4 (empno, ename, job, sal, deptno) select empno, ename, job, sal, deptno from scott.emp;

19. SQL> alter table scott.emp4 rename to test5;

20. The unique key constraint does allow for null values, Primary Key does not. You can only specify one Primary Key column, but you may give multiple columns the Unique constraint.

21. SQL> alter table scott.emp2 drop constraint UK\_EMPID;

22. SQL> alter table scott.product\_master disable constraint PK\_INV;

23. SQL> alter table scott.emp2 add sex varchar2(1);

24. SQL> alter table emp\_test2 modify FIRSTNAME varchar2(30);

SQL> alter table emp\_test2 modify LASTNAME varchar2(30);

25. SQL> create or replace view EMPLOYEES\_VW as select employee\_id, phone\_number, first\_name||' '||last\_name "EMPLOYEE", department\_id from hr.employees;

26. SQL> select employee, department\_id, employee\_id from EMPLOYEES\_VW;

27. SQL> create view DEPTVW as select phone\_number "EMPNO", last\_name "EMPLOYEE", department\_id "DEPTNO" from hr.employees;

28. SQL> create sequence DEPT\_SEQ minvalue 300 start with 300 maxvalue 1000 increment by 10;

SQL> insert into testdepartments (department\_id) values (DEPT\_SEQ.nextval);

29. SQL> create index lname\_idx on scott.emp2(last\_name);

30. SQL> create public synonym scotty for scott.emp;

31. SQL> create index Name on hr.employees(first\_name, last\_name);

32. SQL> drop table scott.product\_master;

33. SQL> create table FACULTY (FID number not null, I\_NAME varchar2(20), I\_LOC varchar2(20), constraint PK\_FID primary key(FID));

34. SQL> create table SPECIALTY (SPID number not null, SP\_DESC varchar2(20), constraint PK\_SPID primary key(SPID));

35. SQL> create table FAC\_SPECIALTY (FID number not null, SPID number not null, constraint fid\_fk foreign key(fid) references FACULTY(FID), constraint spid\_fk foreign key(SPID) references SPECIALTY(SPID));

36. SQL> create table STUDENT (SID number not null, NAME varchar2(20), MAJOR varchar2(20), GRADE\_LEVEL varchar2(10), AGE number not null, ADVISOR number not null, constraint pk\_SID primary key(SID), constraint FK\_ADVISOR foreign key(ADVISOR) references FACULTY(FID));

37. SQL> create table CLASS (C\_NAME varchar2(20), TIME varchar2(7), ROOM varchar2(5), FID number not null, constraint FK\_FID foreign key(FID) references FACULTY(FID), constraint pk\_cname primary key(C\_NAME));

38. SQL> create table ENROLLMENT (SID number not null, C\_NAME varchar2(20), constraint fk\_cname foreign key(C\_NAME) references CLASS(C\_NAME), constraint fk\_sid foreign key(SID) references STUDENT(SID));

39. SQL> insert into FACULTY values (100, 'AKIN', '100BC');

40. SQL> insert into SPECIALTY values (001, 'Firewall');

41. SQL> insert into FAC\_SPECIALTY values (100, 001);

42. SQL> insert into FACULTY values (0, 'JONES', 'NULL');

SQL> insert into STUDENT values (100, 'JONES', 'MIS', 'GR', 35, 0);

43. SQL> insert into CLASS values ('INSS111', '7:15W', '355BC', 200);

44. SQL> insert into CLASS values ('INSS651', 'NULL', 'NULL', 100);

SQL> insert into ENROLLMENT values (100, 'INSS651');

SQL> insert into FACULTY values (350, 'NULL', 'NULL');

SQL> insert into CLASS values ('INSS610', 'NULL', 'NULL', 350);

SQL> insert into ENROLLMENT values (350, 'INSS610');

45. SQL> select I\_NAME, SP\_DESC from FACULTY join FAC\_SPECIALTY using(FID) join SPECIALTY using(SPID);

46. SQL> select \* from STUDENT;

47. SQL> select I\_NAME, SP\_DESC from FACULTY join FAC\_SPECIALTY using(FID) join SPECIALTY using(SPID);

48. SQL> select FID, I\_NAME, SP\_DESC from FACULTY join FAC\_SPECIALTY using(FID) join SPECIALTY using(SPID) where SP\_DESC in ('Marine Reconnaisance');

49. SQL> select NAME, AGE, I\_NAME from STUDENT join ENROLLMENT using(SID) join CLASS using(C\_NAME) join FACULTY using(FID) where I\_NAME = 'TYSON';

50. SQL> select NAME, I\_NAME from STUDENT join ENROLLMENT using(SID) join CLASS using(C\_NAME) join FACULTY using(FID);

51. SQL> create table Student2 as select \* from STUDENT;

SQL> create table Class2 as select \* from CLASS;

SQL> create table Faculty2 as select \* from FACULTY;

52. SQL> drop table Student2;

SQL> drop table Class2;

SQL> drop table Faculty2;

53. SQL> create table test8 as select \* from scott.emp;

54. SQL> update test8 set SAL = 3200, JOB = 'SALESMAN', DEPTNO = 40 where ENAME = 'SMITH';

55. SQL> update test8 set SAL = 72000, JOB = 'BOSS' where ENAME = 'JAMES';

SQL> commit;

56. SQL> delete from test8 where DEPTNO = 20 and JOB not in ('PRESIDENT');

57. SQL> delete from test8 where COMM > 0;

58. SQL> rollback;

59. SQL> delete from test8;

SQL> commit;

60. SQL> rename test8 to test9;

61. SQL> drop table test9;

62. Primary Key is the most powerful constraint, it uniquely identifies each row in a column from each other and doesn't accept duplicate or null values.

63. A Foreign Key is the child column in relation to the Primary Key parent column that resides in a different table.

64. The difference between a primary key and a unique key is that the Unique key allows for null values, PK does not.

65. The primary key/foreign key relationship is Parent Column/Child Column

66. The check constraint ensures that certain criteria are met on a column as data is entered into that column, it makes sure data that is entered meets the specified requirement.

67. SQL> select last\_name, first\_name, department\_name, location\_id from hr.employees join hr.departments using(department\_id);

68. SQL> select last\_name, salary from hr.employees where salary > (select salary from hr.employees where last\_name = 'Chung');

69. SQL> select last\_name, department\_id from hr.employees where department\_id = (select department\_id from hr.employees where last\_name = 'Whalen');

70. SQL> select ename, dname from scott.emp join scott.dept using(deptno);

71. SQL> select empno, job, loc from scott.emp join scott.dept using(deptno);

72. SQL> select \* from scott.salgrade;

73. SQL> select order\_id, product\_id from oe.order\_items;

74. SQL> select unit\_price, quantity, order\_date, order\_mode from oe.order\_items join oe.orders using(order\_id)

75. SQL> l

1\* select line\_item\_id, customer\_id, warehouse\_id from oe.orders join oe.order\_items using(order\_id) join oe.inventories using(product\_id)

76. SQL> l

1\* select sales\_rep\_id, marital\_status, cust\_first\_name, cust\_last\_name from oe.orders join oe.customers using(customer\_id)

77. SQL> l

1\* select warehouse\_spec, warehouse\_name, quantity\_on\_hand from oe.warehouses join oe.inventories using(warehouse\_id)

78. SQL> l

1\* select \* from oe.product\_descriptions

79. SQL> l

1\* select \* from oe.product\_information

80. SQL>

1\* select translated\_name, weight\_class, supplier\_id, product\_status, list\_price, line\_item\_id from oe.product\_descriptions join oe.product\_information using(product\_id) join oe.order\_items using(product\_id)

81. SQL> l

1\* select \* from oe.orders

82. SQL> create view hr.orders\_vxw as select order\_id, order\_date, promotion\_id from oe.orders;

create view hr.orders\_vxw as select order\_id, order\_date, promotion\_id from oe.orders

\*

ERROR at line 1:

ORA-01031: insufficient privileges

83. SQL> create view scott.orders\_vwx2 as select order\_total, line\_item\_id, quantity\_on\_hand from oe.orders join oe.order\_items using(order\_id) join oe.inventories using(product\_id);

create view scott.orders\_vwx2 as select order\_total, line\_item\_id, quantity\_on\_hand from oe.orders join oe.order\_items using(order\_id) join oe.inventories using(product\_id)

\*

ERROR at line 1:

ORA-01031: insufficient privileges

84. SQL> create view sys.warehouse\_vw as select warehouse\_name, product\_id from oe.warehouses join oe.inventories using(warehouse\_id);

View created.

85. SQL> l

1\* select product\_description, translated\_name, translated\_description, weight\_class, warehouse\_id from oe.product\_information join oe.product\_descriptions using(product\_id) join oe.inventories using(product\_id)

86. (can't drop view that isn't made)

87. (can't drop view that isn't made)

88. SQL> delete from sys.warehouse\_vw;

1112 rows deleted.

89. SQL> rollback;

90. SQL> drop view warehouse\_vw;

91. SQL> create table warehouse\_new as select translated\_name, weight\_class, supplier\_id, product\_status, list\_price, line\_item\_id from oe.product\_descriptions join oe.product\_information using(product\_id) join oe.order\_items using(product\_id);

92. SQL> l

1\* select \* from warehouse\_new

93. SQL> create table hr.orders\_new as select sales\_rep\_id, marital\_status, cust\_first\_name, cust\_last\_name from oe.orders join oe.customers using(customer\_id);

94. SQL> select \* from hr.orders\_new;

95. SQL> delete from hr.orders\_new;

96. SQL> rollback;

97. SQL> truncate table hr.orders\_new;

98. SQL> drop table hr.orders\_new purge;

99. SQL> select \* from hr.orders\_new;

select \* from hr.orders\_new

\*

ERROR at line 1:

ORA-00942: table or view does not exist

100. SQL> create table oe.employees\_new as select last\_name, department\_name, city, phone\_number, postal\_code from hr.employees join hr.departments using(department\_id) join hr.locations using(location\_id);

101. SQL> drop table oe.employees\_new purge;