1. **Create**the two tables, **Worker** and **Skill**, with data types as befitting the data you see in the sample data. Do not define the primary key for each table at this time. Do not insert data at Q1 otherwise you won’t be able to complete Q3 correctly.  Use multiple SQL Statements.

CREATE TABLE Worker( WorkerId Number, LastName VarChar2(25), FirstName Varchar2(10), SkillId varchar(5));

CREATE TABLE Skill( SkillId varchar(5), descrip VarChar2(25), HourlyMax Number, ApprovedBy Number);

1. Use multiple SQL statements to define the following primary keys and foreign keys for those two tables. Think carefully about the sequence of SQL statements used here. Submit the SQL statements in the right sequence. (Hint: A foreign key references the primary key of another table, which means the referenced primary key needs to be defined before you can define the foreign key.)

ALTER TABLE Worker ADD CONSTRAINT Worker\_WorkerId\_pk PRIMARY KEY(WorkerId);

ALTER TABLE Skill ADD CONSTRAINT Skill\_SkillId\_pk PRIMARY KEY(SkillId);

1. Without dropping any of the constraints already defined on both Worker and Skill tables, insert the data shown in the Skill and Worker tables (2 rows for worker and 2 rows for skill). Keep in mind that database enforces referential integrity rules so you can’t put in data that would violate any referential integrity rules. **You must use ONLY DML statements (multiple needed) to accomplish the task required in this question, that is, do not drop foreign key constraints (which would require a DDL).** Submit all relevant DML statements used.

INSERT ALL INTO Skill (skill.skillid, skill.descrip, skill.hourlymax) VALUES('K1', 'COMPUTING', 50)

INTO Worker(WorkerId, LastName, FirstName, skillId) VALUES(1001, 'PARKER', 'AMY', 'K1')

INTO Worker(WorkerId, LastName, FirstName) VALUES(1002, 'SMITH', 'JANE')

INTO Skill (skill.skillid, skill.descrip, skill.hourlymax, skill.approvedby) VALUES('K2', 'MANAGEMENT', 80, 1002)

SELECT \* FROM dual;

update worker set skillid = 'K1' where workerid = 1002;

update skill set approvedby = 1002 where skillId = 'K1';

1. Define that HourlyMax is 15 or higher.

ALTER TABLE skill

ADD CONSTRAINT skill\_HourlyMax\_ck CHECK (HourlyMax>=15);

1. Add a column to Worker Table. Column name is Hiredate.

ALTER TABLE Worker ADD Hiredate Date;

1. Enter 12/16/2019 as the hiredate for Worker 1001. Enter today’s date without time (i.e. time would be 00:00:00) as the hiredate for Worker 1002.  You're not limited to one SQL statement for this task.

update worker set hiredate = TO\_DATE('12/16/2019', 'mm/dd/yyyy') where workerid = 1001;

update worker set hiredate = TO\_DATE(SYSDATE, 'mm/dd/yyyy') where workerid = 1002;

1. Use one SQL to accomplish this: change skill.descrip to lower case for all rows and increase all hourly max by 10%. You’ll need to change the data in the table, not just viewing descrip in lower case and not just viewing the new hourly max.

update skill set descrip = lower(descrip), hourlymax = hourlymax + (hourlymax \* 0.1);

1. Find all **check constraints** defined on emp. In the result, list the name of the constraint and the actual restriction defined (such as ‘sal>9999’) (Search\_condition column).

select constraint\_name, table\_name, constraint\_type, search\_condition

From user\_constraints

Where table\_name='EMP' and constraint\_type = 'C';

1. Find all **foreign key** constraints defined on emp table and on dept table. In the list, list the name of the constraint and which table it is defined on. USE only one SQL statement to accomplish this.

select constraint\_name, table\_name, constraint\_type

From user\_constraints

Where table\_name in ('EMP', 'DEPT') and constraint\_type = 'R';

1. Give a list of tables that have foreign keys.

select constraint\_name, table\_name, constraint\_type

From user\_constraints

Where constraint\_type = 'R';

1. Give a list of tables that do not have any foreign keys.

select distinct table\_name From user\_constraints Where table\_name not in (select distinct table\_name From user\_constraints Where constraint\_type = 'R');

1. Find the name of Turner’s manager.

select empno, ename from emp where empno = (select mgr from emp where ename = 'TURNER');

1. Find a list of customers (show custid) who **did not** place an order during the first **three** quarters of 2016. (There are 5 such customers: 101, 103, 105, 107, and 108.) You can use common knowledge concerning dates for each quarter. (For example, the first quarter is Jan. 1 to March 31.)

select custid from customer where custid not in (select distinct custid from ord where orderdate between '01-Jan-2016' and '30-Sep-2016');

1. Find out which order has the highest total. In the output, show this order id only.

select ordid from ord where total = (select max(total) from ord)