Student:

**Instructions:**

* In this exercise, you will run some SQL statements to modify primary and foreign key data values and analyze the outcome of the statements.
* If a statement runs correctly, use the SELECT command to observe the change in the tables.
* If a statement causes an error, explain the reason of the error.
* At the end of exercise, you are expected to come up with a set of general rules for the behavior of referential integrity rules.

**Step 1.** Copy/Paste the SQL script into your database account and run it.

drop table if exists WORKDONE ;

drop table if exists JOB;

drop table if exists WORKER;

create table WORKER (

TaxID INT ,

FirstName CHAR(30) ,

LastName CHAR(30) ,

primary key (TaxID)

);

create table JOB (

ID INT,

Description CHAR(30),

Rate FLOAT,

primary key(ID)

);

create table WORKDONE (

TNo INT,

CheckIn DATETIME,

CheckOut DATETIME,

WORKER\_TaxID INT,

JOB\_ID INT,

primary key(TNo),

foreign key(WORKER\_TaxID) references WORKER(TaxID)

on update cascade on delete cascade,

foreign key(JOB\_ID) references JOB(ID)

on update restrict on delete restrict

);

/\*allow unsafe updates\*/

SET SQL\_SAFE\_UPDATES = 0;

/\*data\*/

insert into JOB values

(1,"Help Desk",7.00),

(2,"Lab Assistant",8.50),

(3,"PC Technician",9.50),

(4,"Audio/Visual",8.50),

(5,"Night Support",9.00);

insert into WORKER values

(1, 'Betty', 'Missing'),

(2, 'Bob', 'Slacker'),

(3, 'John', 'Doe'),

(4, 'Joe', 'Noanswer');

insert into WORKDONE values

(1,'2017-09-20 10:00','2017-09-20 11:00',1,2),

(2,'2017-09-20 11:00','2017-09-20 12:00',3,2),

(3,'2017-09-20 23:00','2017-09-20 00:00',4,1);

**Step 2.** Open new Query Window () and use select \* from *tablename* to make sure that all tables are created with data. Analyze the rows of the tables. Identify the parent tables, child tables, and primary-foreign key relationships between the tables.

**The parent tables are the Worker and Job tables and that would make the child table the work done table. The primary key in the worker table is the TaxID number which makes it the foreign key in the WorkDone table. The primary key in the Job table is the Job ID and it is also a foreign key in the WorkDone table. The WorkDone table’s primary key is TNo.**

|  |  |
| --- | --- |
| **WORKER** | **JOB** |
| **WORKDONE** |  |

**Step 3.** Run the SQL query examples given below and observe their impact on the child table WORKDONE. Briefly explain what happens and why? Justify your answers based on the lecture notes (Section 3.1 to 3.3 Creating Databases with SQL)

|  |  |  |  |
| --- | --- | --- | --- |
| **Action in the Parent** | **Referential Integrity Rule in MYSQL** | **Example** | **Outcome in the child table (WORKDONE) and parent table**  **Reflection-Why does the example run or not run?** |
| Try to change a job id in the JOB table | ON UPDATE RESTRICT | update job  set id=10  where id=1; | It does run but it does not change anything because of you can’t delete a foreign key constraint. |
| Try to change the ID from 50 to 5 in the JOB table | ON  UPDATE  RESTRICT | update job  set id=50  where id=5; | It does run because the order of the stamen is correct and the number 5 is an ID number in the JOB table. |
| Try to change a worker id in the WORKER table | ON UPDATE CASCADE | update worker  set TaxId=10  where TaxId=1; | It does run and it changed the TaxID from 1 to 10. It also moves the 10 ID down so it will be in chronological order. |
| Try to delete a job in the JOB table | ON DELETE RESTRICT | delete from job  where id=2; | It will not run because of the relationships between the JOB table and the other tables because it would delete the foreign keys in those tables. |
| Try to delete from the ID from job table. | ON DELTE  CASCADE | delete from job  where id=3; | It does run then deletes the id number three from the job table. |
| Try to delete a worker in the WORKER table | ON DELETE CASCADE | delete from worker  where TaxId=3; | It does run correctly and it deletes the tax id three from the table Worker. |

**Step 4.**

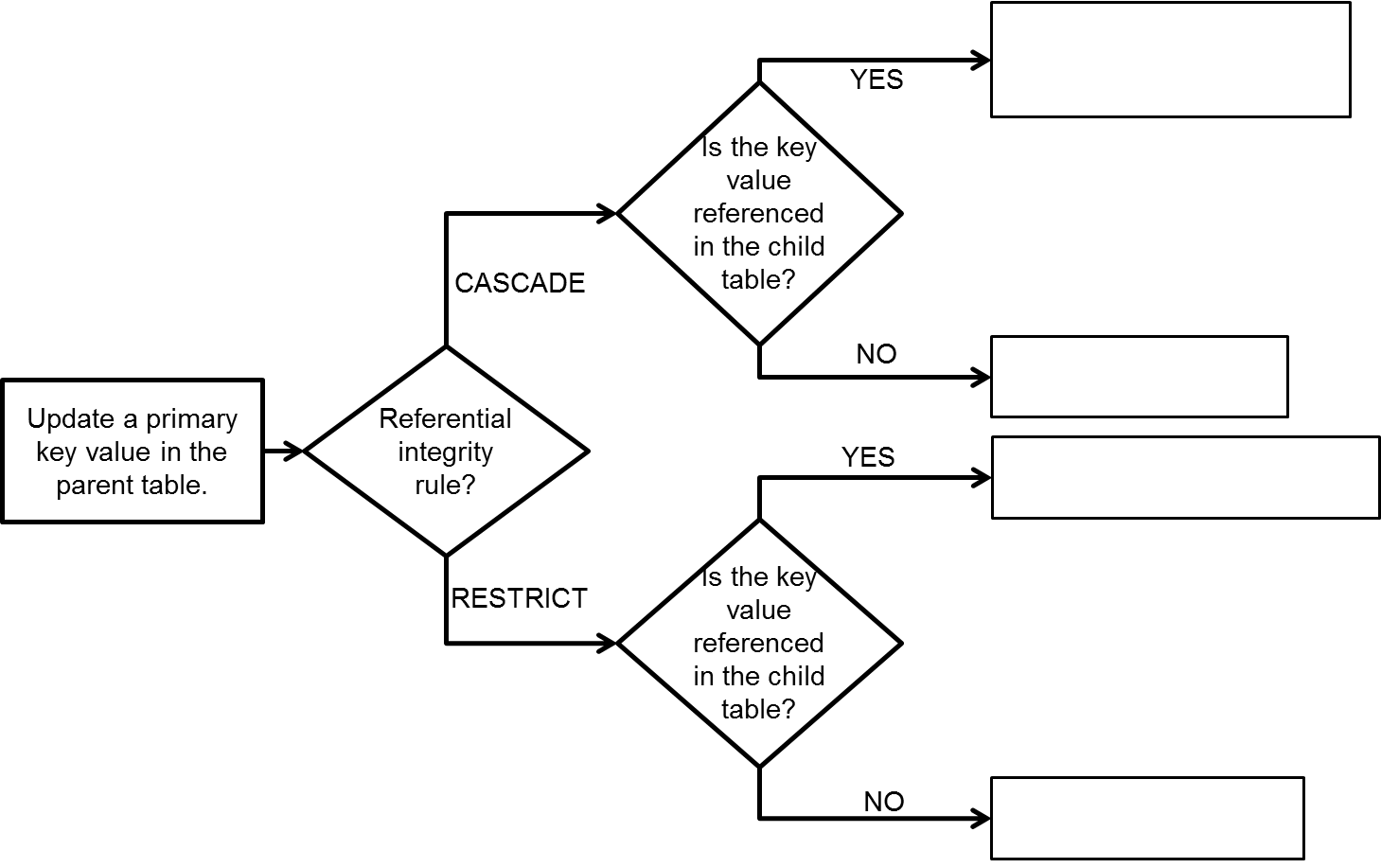
Run the SQL query examples given below and observe their impact on the parent tables.

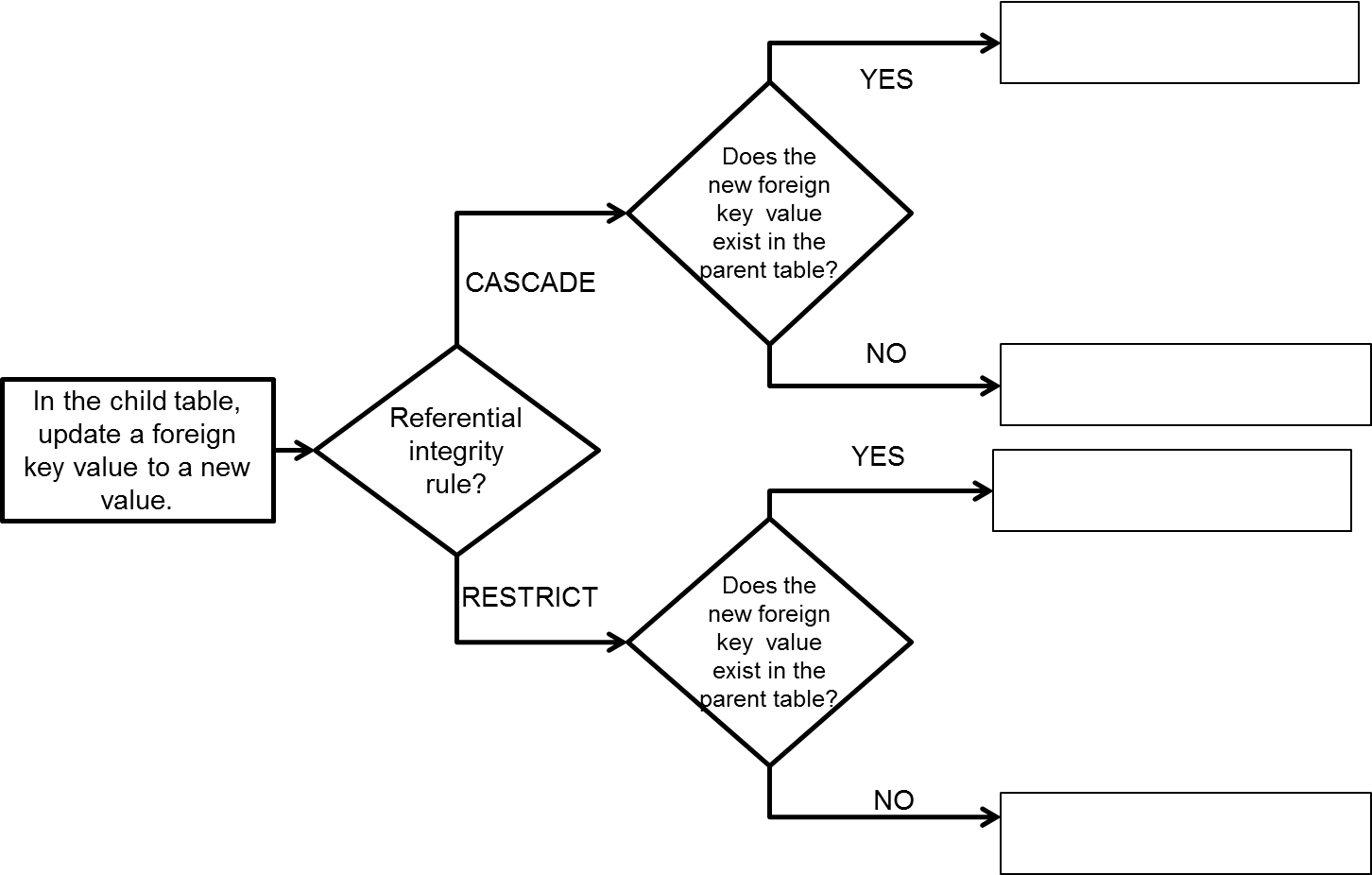
|  |  |  |  |
| --- | --- | --- | --- |
| **Action in the child** | **Referential Integrity Rule in MYSQL** | **Example** | **Outcome in the child table (WORKDONE) and parent table**  **Reflection-Why does the example run or not run?** |
| Try to change a job id in the WORKDONE table | ON UPDATE RESTRICT | update workdone  set JOB\_id=20  where JOB\_id=2; | This statement does not run because you can’t update the child table without updating the parent table. |
| Trying to change the job id in the WORKDONE table | ON  UPDATE | update workdone  set JOB\_id=1  where JOB\_id=2; | This statement does work and it sets the all Job ids that where two to the number one. |
| Try to change a worker id in the WORKDONE table | ON UPDATE CASCADE | update workdone  set WORKER\_TaxId=40  where WORKER\_TaxID=4; | This statement does not work because of the child table not being able to change without the parent table changing. |
| Try to change the taxid in the WORKDONE table | ON  UPDATE  RETRICT | update workdone  set WORKER\_TaxId=2  where WORKER\_TaxID=4; | This statement does run correctly and it changes the TaxId in the worker table from two to 4. |
| Try to delete rows in the WORKDONE | ON DELETE CASCADE | delete from workdone; | This statement does run correctly and it deletes all the rows from the WORKDONE table. |

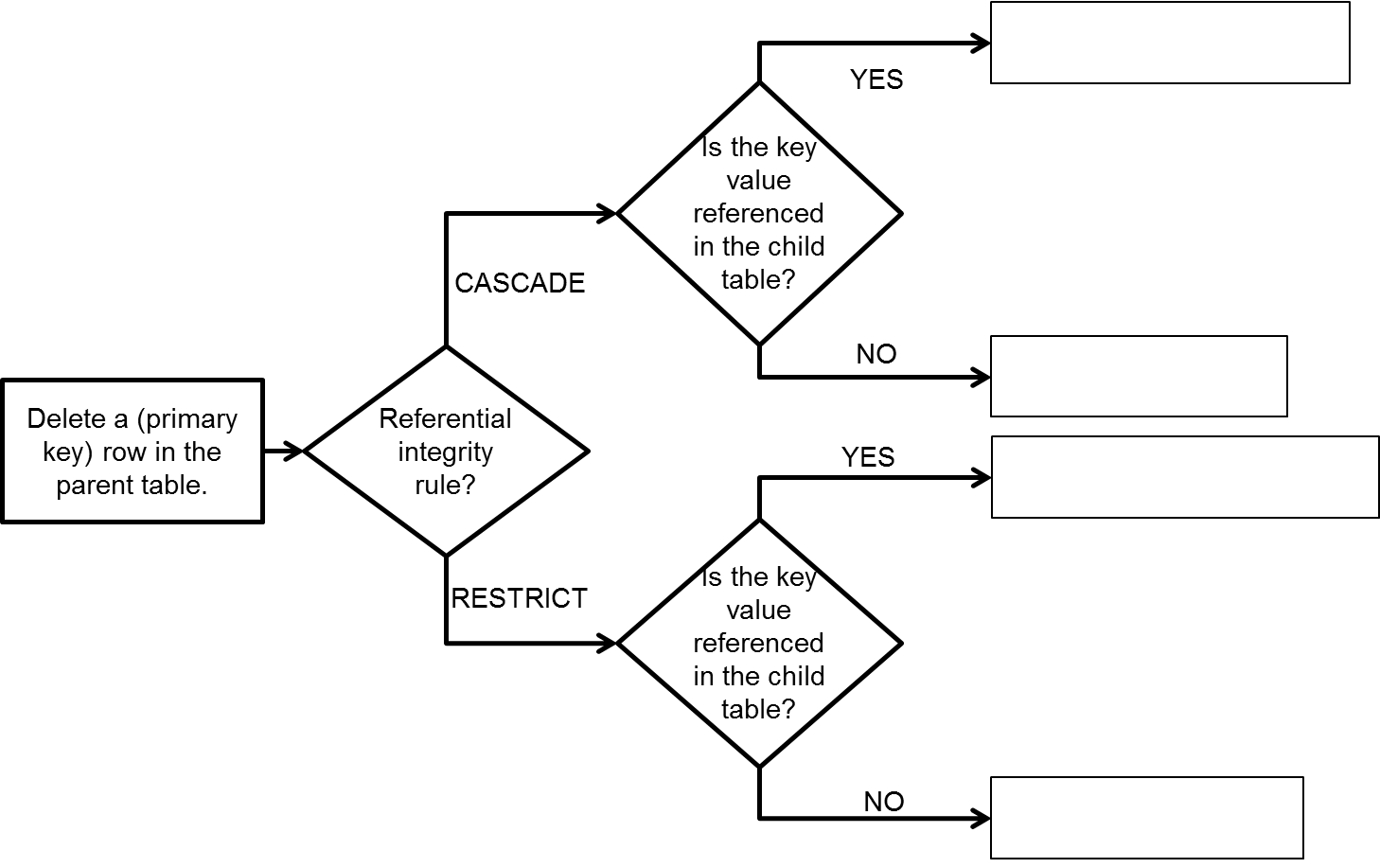
Step 5 Answers:

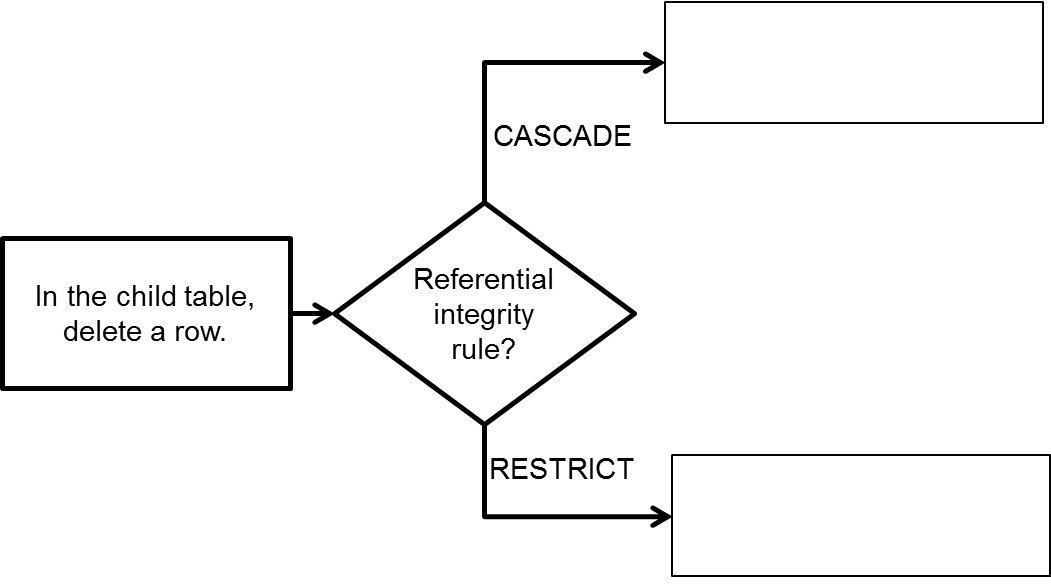
1. All the following rows are updated
2. No changes
3. Error
4. The update can be done in the parent table only
5. The update can run correctly
6. A error occurs
7. The update does run

**Step 5. No submission required.** Based on the results of the experiments above, come with general rules explaining how referential integrity rules work in different cases. Fill in the boxes (Error, No Change, or Explain the action). This is a test practice for you. You are not expected to complete this part in your lab report.









Step 5 answers Continued

1. An error occurs
2. All of the rows in both tables are deleted
3. Only rows in the parent table are deleted
4. An error occurs.
5. Doesn’t change anything in the parent table but a child row is deleted

**Step 6. No submission required**. Answer the following questions as a test practice. You are not expected to complete this part in your lab report.

|  |  |  |
| --- | --- | --- |
| **Action / Outcome**  **Parent** | **Referential Integrity Rule in MYSQL** | **Action /Outcome**  **Child** |
| In the parent table, we update a key value that is referenced by the foreign keys of the child table. | ON UPDATE CASCADE | The key value in the child table changes and then is updated to the new value. |
| In the parent table, we update a key value that is referenced by the foreign keys of the child table. | ON UPDATE NO ACTION /  ONUP DATE RESTRICT | The value changes in the foreign child table because it was changed in the parent table. |
| In the parent table, we delete a row that is referenced by the foreign keys of the child table. | ON DELETE CASCADE | In the parent table and child table the corresponding row will be deleted. |
| In the parent table, we delete a row that is referenced by the foreign keys of the child table. | ON DELETE NO ACTION /  ON DELETE RESTRICT | The row in the parent and the child tables are deleted. |
| In the child table we try to update the foreign key in the child table which is also the primary key in the parent table. | ON UPDATE CASCADE  ON UPDATE NO ACTION  ON UPDATE RESTRICT | In the child table, we update a foreign key to a value that references a primary key value of the parent table. |
| In the parent table nothing is changed because the value being changed doesn’t represent anything in the parent table. | ON UPDATE CASCADE  ON UPDATE NO ACTION  ON UPDATE RESTRICT | In the child table, we update a foreign key to a value that does not reference a primary key value of the parent table. |
| A row in the child table is deleted with no effect on the parent table. | ON DELETE CASCADE  ON DELETE NO ACTION  ON DELETE NO RESTRICT | In the child table, we delete a row. |