

Chapter 8

Review Questions

1. What type of integrity is enforced when a primary key is declared?

Entity Integrity (i.e. PK must be unique and not NULL).

4. What are “referential constraint actions”?

A referential constraint action is an insertion, update, or deletion action in the referenced table it is specified with CASCADE or RESTRICT to ensure the preservation of referential integrity.

5. What is the purpose of a CHECK constraint?

The CHECK constraint is used to validate data when an attribute value is entered. It checks to see that a specified condition exists. If the CHECK constraint is met for the specified attribute (i.e., the condition is true), the data is accepted for that attribute. If the condition is found to be false, an error message is generated, and the data is not accepted.

6. Explain when an ALTER TABLE command might be needed.

All changes in the table structure are made by using the ALTER TABLE command followed by a keyword that produces the specific change you want to make. Three options are available: ADD, MODIFY, and DROP. For example, the ALTER TABLE command might be needed when: Changing a Column's Datatype, Data Characteristics, Adding/Dropping a Column, Adding Primary Key, Foreign Key, and Check Constraints.

7. What is the difference between an INSERT command and an UPDATE command?

SQL requires the use of the insert command to enter data into a table while UPDATE command is used to modify data in a table. INSERT creates new rows in the table, while UPDATE changes rows that already exist.

9. What is the difference between a view and a materialized view?

Views can be created to expose subsets of data to end users primarily for security and privacy reasons. Normally, views only store the SELECT statement to produce the view. Materialized views store a separate copy of the data and must be refreshed regularly. A materialized view is a dynamic table that not only contains the SQL query command to generate the rows, it stores the actual rows. The materialized view rows are automatically updated when the base tables are updated

Problems

1. Write the SQL code that will create only the table structure for a table named EMP_1. This table will be a subset of the EMPLOYEE table. The basic EMP_1 table structure is summarized in the following table. Use EMP_NUM as the primary key. Note that the JOB_CODE is the FK to JOB so be certain to enforce referential integrity. Your code should also prevent null entries in EMP_LNAME and EMP_FNAME.

```
CREATE TABLE EMP_1 (  
    EMP_NUM CHAR(3) NOT NULL,  
    EMP_LNAME VARCHAR(15) NOT NULL,  
    EMP_FNAME VARCHAR(15) NOT NULL,  
    EMP_INITIAL CHAR(1) NULL DEFAULT NULL,  
    EMP_HIREDATE DATE NULL DEFAULT NULL,  
    JOB_CODE CHAR(3) NULL DEFAULT NULL,  
    PRIMARY KEY (EMP_NUM),  
    FOREIGN KEY (JOB_CODE) REFERENCES JOB(JOB_CODE) ON UPDATE  
    CASCADE  
);
```

4. Write the SQL code that will save the changes made to the EMP_1 table (if supported by your DBMS).

```
COMMIT;
```

7. Write the SQL code to create a copy of EMP_1, including all its data, and naming the copy EMP_2.

```
CREATE TABLE EMP_2 AS  
SELECT * FROM EMP_1;
```

10. Using the EMP_2 table, write a single SQL command to change the EMP_PCT value to 5.00 for the people with employee numbers 101, 105, and 107.

```
UPDATE EMP_2  
SET EMP_PCT = 5.00  
WHERE EMP_NUM IN (101, 105, 107);
```