Aman sir ct solution(Male section)

1)Write down the difference between integrity and security system.

Ans:

S.No.	Data Security	Data Integrity
1.	Data security refers to the prevention of data corruption through the use of controlled access mechanisms.	Data integrity refers to the quality of data, which assures the data is complete and has a whole structure.
2.	Its motive is the protection of data.	Its motive is the validity of data.
3.	Its work is to only the people who should have access to the data are the only ones who can access the data.	Its work is to check the data is correct and not corrupt.
4.	It refers to making sure that data is accessed by its intended users, thus ensuring the privacy and protection of data.	It refers to the structure of the data and how it matches the schema of the database.
5.	Some of the popular means of data security are authentication/authorization, masking, and encryptions.	Some of the means to preserve integrity are backing up, error detection, designing a suitable user interface and correcting data.
6.	It relates to the physical form of data against accidental or intentional loss or misuse and destruction.	It relates to the logical protection (correct, complete and consistence) of data.
7.	It avoids unauthorized access of data.	It avoids human error when data is entered.
8.	It can be implemented through : • user accounts (passwords) • authentication schemes	It can be implemented by following rule : • Primary Key • Foreign Key • Relationship

2)Student(id,name,gpa,semester,fee)
write an assertion allows discounted payment of fees if the his/her gpa is more than 3.75 in consecutive semester
Ans:

```
CREATE ASSERTION GPAConsecutiveSemesters
CHECK (
  NOT EXISTS (
    SELECT 1
    FROM Student s1
    WHERE s1.gpa > 3.75
      AND s1.semester = (SELECT MAX(semester) FROM Student s2 WHERE s2.id = s1.id)
  )
);
```

Aman sir ct solution(Female section)

show the importance of data integrity constraints and write down the example

Data integrity constraints are rules or conditions that are defined in a database to ensure the accuracy, consistency, and reliability of the data. They play a crucial role in maintaining data integrity by preventing the entry of incorrect or inconsistent data. The importance of data integrity constraints can be highlighted through the following key points:

1. **Data Accuracy**: Data integrity constraints help ensure that the data stored in a database is accurate and reliable. By enforcing rules and constraints, errors and inconsistencies are minimized, leading to trustworthy data.
2. **Data Consistency**: Constraints maintain the consistency of data within the database. They prevent situations where data in one part of the database contradicts data in another part, which can lead to confusion and incorrect decision-making.
3. **Preventing Data Anomalies**: Constraints can prevent data anomalies such as insertion, update, or deletion anomalies, which can occur when data is not properly controlled. Anomalies can lead to data corruption and issues in data retrieval and analysis.
4. **Data Relationships**: In relational databases, constraints help maintain the relationships between tables. Foreign key constraints, for example, ensure that records in one table correspond to records in another, helping to maintain referential integrity.
5. **Enforcing Business Rules**: Data integrity constraints enable the enforcement of specific business rules and requirements. This ensures that the data stored in the database aligns with the organization's standards and practices.
6. **Security**: Constraints can also contribute to data security by preventing unauthorized or malicious changes to the data. For example, a CHECK constraint can prevent the insertion of invalid data.
7. **Data Quality**: Data quality is significantly improved with data integrity constraints in place. High-quality data is essential for informed decision-making and reliable reporting.
8. **Error Reduction**: Constraints help reduce human errors in data entry. They serve as a safety net by automatically rejecting data that doesn't meet the defined criteria.

Example of Data Integrity Constraints:

Let's consider a simple example in the context of a database for an online store. The database contains two tables: `Products` and `Orders`. Data integrity constraints are used to maintain data accuracy and consistency.

1. **Primary Key Constraint**: In the `Products` table, the `product_id` column is defined as the primary key. This constraint ensures that each product has a unique identifier. Attempting to insert a duplicate `product_id` would result in an error, preventing data duplication.
2. **Foreign Key Constraint**: In the `Orders` table, the `product_id` column references the `Products` table. This is a foreign key constraint that ensures that orders can only be placed for products that exist in the `Products` table, maintaining referential integrity.
3. **Check Constraint**: A check constraint can be used to ensure that the price of a product is always greater than or equal to zero. This prevents the insertion of products with a negative price.
4. **Unique Constraint**: In the `Orders` table, a unique constraint can be applied to the combination of `order_id` and `product_id`. This ensures that a specific product can only appear once in a single order.

By implementing these data integrity constraints, the database for the online store can maintain accurate and consistent data, prevent errors, and enforce business rules, ultimately contributing to the overall success and reliability of the system.

In Oracle, you can create a trigger to apply a 20% discount to a student's fee when their GPA is updated to 3.75. Here's how you can create this trigger in Oracle:

Assuming you have a table named `Student` with the columns `id`, `name`, `gpa`, `semester`, and `fee`, you can use the following PL/SQL code to create the trigger:

2)Student(id,name,gpa,semester,fee)
CREAT A TRIGGER WHERE NEW DISCOUNT FEE will be 20% If updated GPA is 3.75

ANS:

```
CREATE OR REPLACE TRIGGER
ApplyDiscountOnGPAUpdate
AFTER UPDATE ON Student
FOR EACH ROW
BEGIN
  IF :NEW.gpa = 3.75 AND :OLD.gpa <> 3.75 THEN
    :NEW.fee := :NEW.fee * 0.8; -- Apply a 20%
discount
  END IF;
END;
/
```

In this Oracle trigger:

- `AFTER UPDATE` specifies that the trigger should execute after an update operation on the `Student` table.
- `FOR EACH ROW` indicates that the trigger should execute once for each row that is updated.
- The trigger checks if the `:NEW.gpa` (the newly updated GPA) is 3.75 and the `:OLD.gpa` (the previous GPA) is not 3.75. This condition ensures that the GPA was updated to 3.75.
- If the condition is met, it applies a 20% discount to the `:NEW.fee` by multiplying it by 0.8, effectively reducing the fee by 20%.

You can run this SQL code in Oracle SQL*Plus or any other Oracle SQL command interface, provided that you have the necessary permissions to create triggers in your Oracle database.

Please make sure to adapt the code and constraints according to your specific database structure and requirements.

manisha