

Previous Question

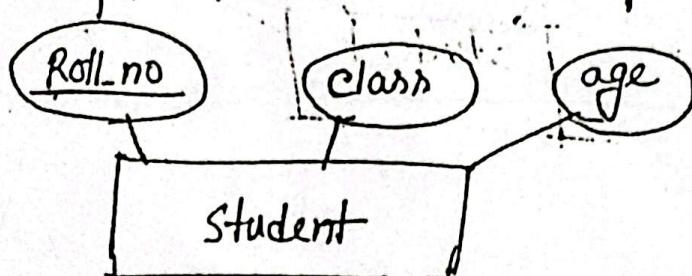
- (Q) Explain the different types of attributes with an appropriate example. ✓

There are six types of attributes. These are

1. Simple attribute
2. composite attribute
3. single valued attribute
4. multivalued attribute
5. derived attribute
6. key attribute

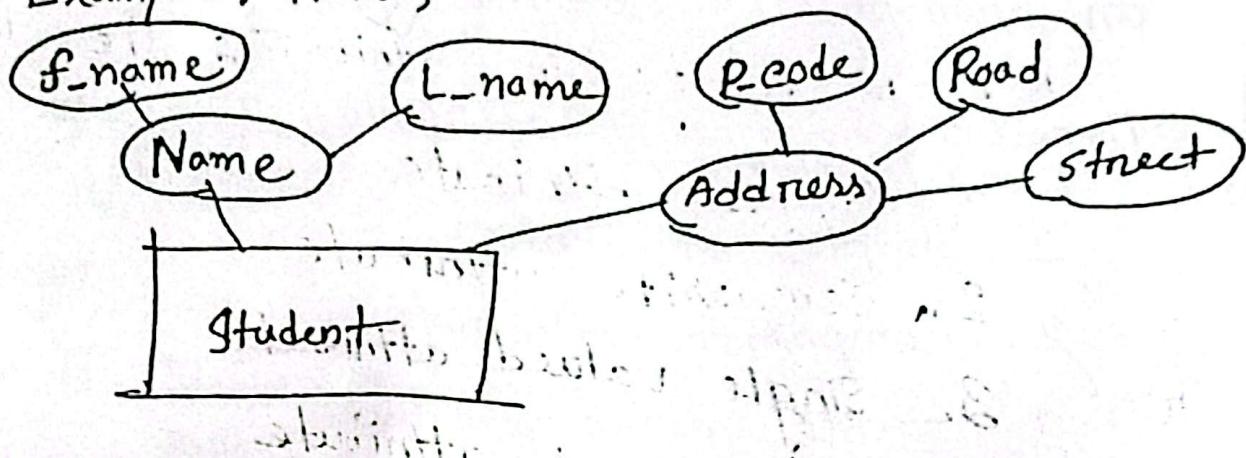
Simple: Simple attribute are those attributes which can not be divided further.

Example: Roll-no, class, age



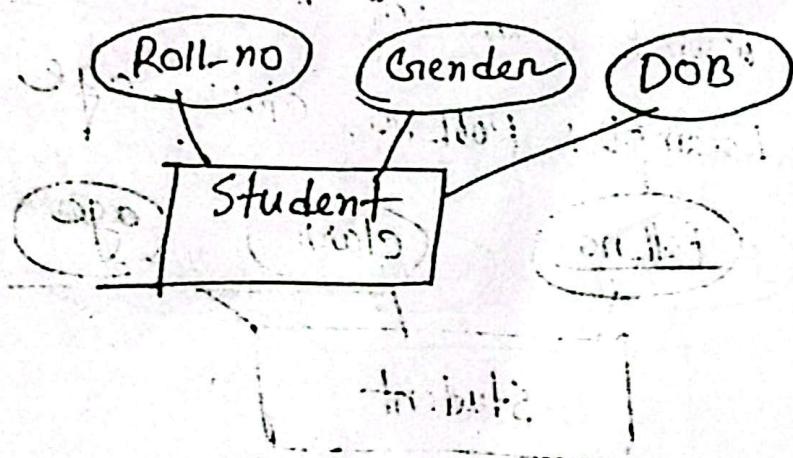
Composite: composite attributes are those attributes composed of many other simple attributes which can be divided into subparts.

Example: Name, address



Single valued: single value attributes are those attributes which can take only one value for given entity from an entity set.

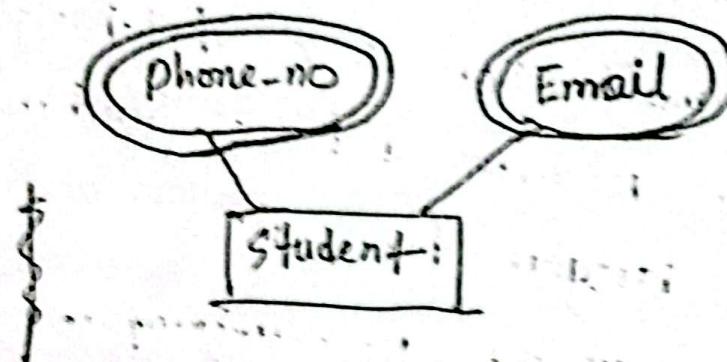
Example: Roll-no, Gender, DOB



Multivalued attribute:

Multivalued attribute are those attributes which can take more than one value for given entity from an entity set.

Example: Phone-no, Email



Derived attribute:

Derived attribute are those attribute which can be derived from other attribute.

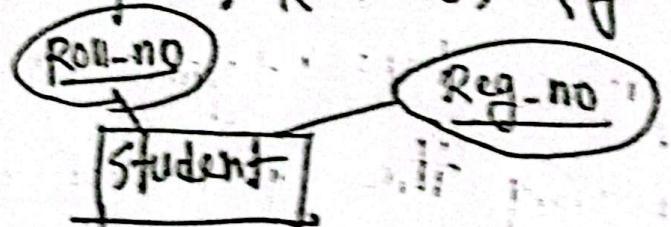
Example: Age (derived from DOB)



Key attribute:

Key attribute are those attributes which can identify an entity uniquely in an entity set.

Example : Roll-no, Reg-no



What is the significance of normalization in database design? Describe different types of normalization with appropriate example.

Normalization:

Normalization is a database design technique without losing information that reduces data redundancy by divides the larger tables into smaller and links them using relationships. Normalization is a database design technique that removes or reduces

Significance : The data & redundancy from a relation.

1. Reduce Data Redundancy
2. No loss of information
3. Save database Space
4. ensure that the table only data contains data directly related to the primary key.
5. Improve data integrity and consistency
6. Increasing the performance.

Sat / Sun / Mon / Tue / Wed / Thu / Fri
□ □ □ □ □ □ □

Date _____

Types of normalization:

There are six types of normalization, but basically we use 3 forms.

1. 1st NF → Remove repeating groups of data
2. 2nd NF → Remove partial dependency
3. 3rd NF → Remove transitive dependency

q. BCNF
5. 4NF

1st NF:

Remove repeating groups of data.

Steps to reduce the repeating groups of data:

(i) pick a value for the repeating data

such (ii) To uniquely identifies each line
within the document.

(iii) we have chosen the Modulecode.

Course → CSC
Module → JAVAP
Database → Database

unnormalized data

Student-course table	
PK	StudentID
	Student Name
	Course Code
	Course Title
	Module Code
	Module Title
	No of credits
	Result code
	Result

1st NF
→

Student-Module-Result	
PK	StudentID
	module code
	Module Title
	No of Credits
	Result Code
	Result

Student course	
PK	StudentID
	Student Name
	Course Code
	Course Title

Module code is the PK for this repeating data.

2nd NF:

Remove Partial dependency

steps of 2nd NF:

(i) you only check for partial dependencies.

(ii) if you have an entity (table) which has a composite key (multi-attribute primary key)

student; course is fully partial independence.
so, we don't change the relation.

Student Module Result	
PK	StudentID
PK	ModuleCode
	ModuleTitle
	No of Credits
	Result code
	Result

Module	
pk	ModuleCode
	ModuleTitle
	No. of Credits

Student-Module-Result	
PK	StudentID
PK	ModuleCode
	ResultCode
	Result

3rd NF:

Remove transitive dependency:

Non-key attr with non-key attr dependency.

Student-course	
PK	StudentID
	StudentName
	CourseCode
	CourseTitle

3rd NF

Course	
PK	courseCode
	courseTitle

Student-course	
PK	Student ID
	Student Name
	Course Code



~~Stt~~

Student-module-result	
PK	StudentID
PK	ModuleCode
	Result Code
	Result

3rd NF

Student-module-result	
PK	StudentID
PK	ModuleCode
	Result code

Result	
PK	Resultcode
	Result

(3) Define various types of keys with appropriate examples and distinguish them from one another.

There are Seven types of Key:

Primary Key:

Primary key is the key that can uniquely identify a tuples or records and cannot contain null values.

Example:

Student

ID	sec	Year

Ex: Reg-id

Super Key:

A super key is a single key or group of key that can uniquely identify the tuples in a table.

Example: Student-NO, (Student-No, name)

Ex: Reg-id, E-mail, phone-no,

sec.Name+Reg-id (can contain redundant attribute)

Age + Reg-id

tuple (Row)

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date / /

Candidate Key:

The minimal set of attributes that can uniquely identify a tuple is known as candidate key.

Candidate keys are a subset of super key.

Example: Reg-id, E-mail-id, phone-no

[Candidate key can contain null values]

Alternate Key:

The Candidate Key other than the primary key is called an alternate key.

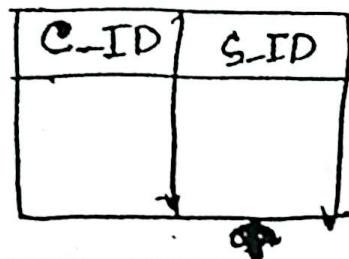
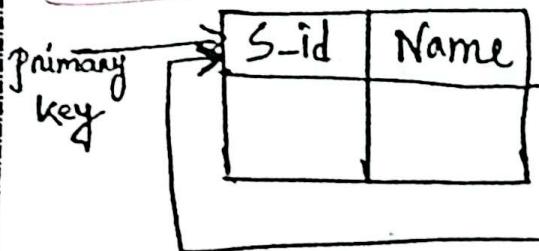
Reg-id, E-mail-id, phone-no are candidate key

Reg-id is primary key so,

E-mail-id and phone-no are Alternate key

Foreign Key:

Foreign key is an attribute or set of attribute in a table that refers to the primary key of another table.



Foreign key

Composite Key:

Composite key is a combination of more than one attribute that can be used to uniquely identify each record.

Ex: (Branch-Name, Branch-Code)

Unique-Key:

Unique key is a candidate key that is not primary key of the relation.

It can accept only null value.

Primary key and unique key have following properties:

Primary Key	Unique Key
Can't be null	Can be null

Primary Key	Unique Key
Can't be repeated	Can be repeated

Q) Why and in which case, triggers can be used?
What system privileges are required to create
a trigger on a table?

Triggers in database management system are special types of stored procedures that automatically execute in response to certain ~~events~~ changes to the data in a table.

Triggers can be useful for enforcing business rules, auditing changes to data, and maintaining complex relationships between tables.

1. Enforcing referential integrity constraints:

For example, when a new row inserted into a child table, a trigger can be used to check if the corresponding parent row exists in the parent table.

2. Auditing changes: A trigger can be used to track all changes to a table in an audit trail table.

3. Implementing complex business rules: For example, a trigger can be used to update a denormalized column in a table whenever related data in another table is updated.

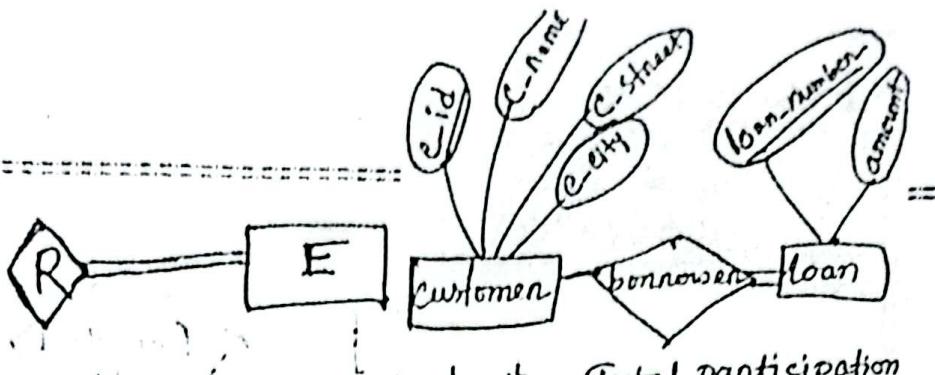
To create a trigger on a table, you need to have the 'CREATE TRIGGER' system privilege. Additionally, you must have the appropriate privileges on the table on which the trigger is being created. These include 'SELECT, INSERT, UPDATE, and DELETE' privileges, depending on the actions that the trigger is designed to handle.

- ⑤ Specify the total and partial participation in the database management system seen.

Total participation:

Total participation refers to a relationship between two entities in which one entity is associated with ~~exactly~~^{at least} one instance of another entity. That is why it also called a mandatory participate.

Total participation is represented using a double line between the entity set & relationship set.



Example:

loan Total participation
customer loan (M2)
borrower | customer partial

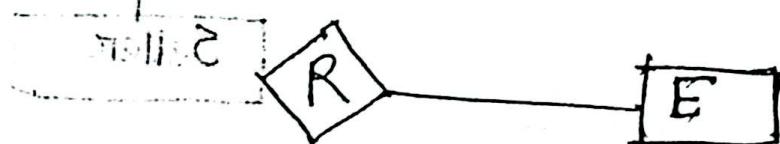


Here double line between the entity set "Student" and relationship set "enrolled in" significant participation. It specifies that each student must be enrolled in at least one course.

Partial participation:

partial participation refers to a relationship in which an entity may or may not be associated with another entity. That is why it is called as optional participation.

partial participation is represented using a single line between the entity set and relationship set.



Example:

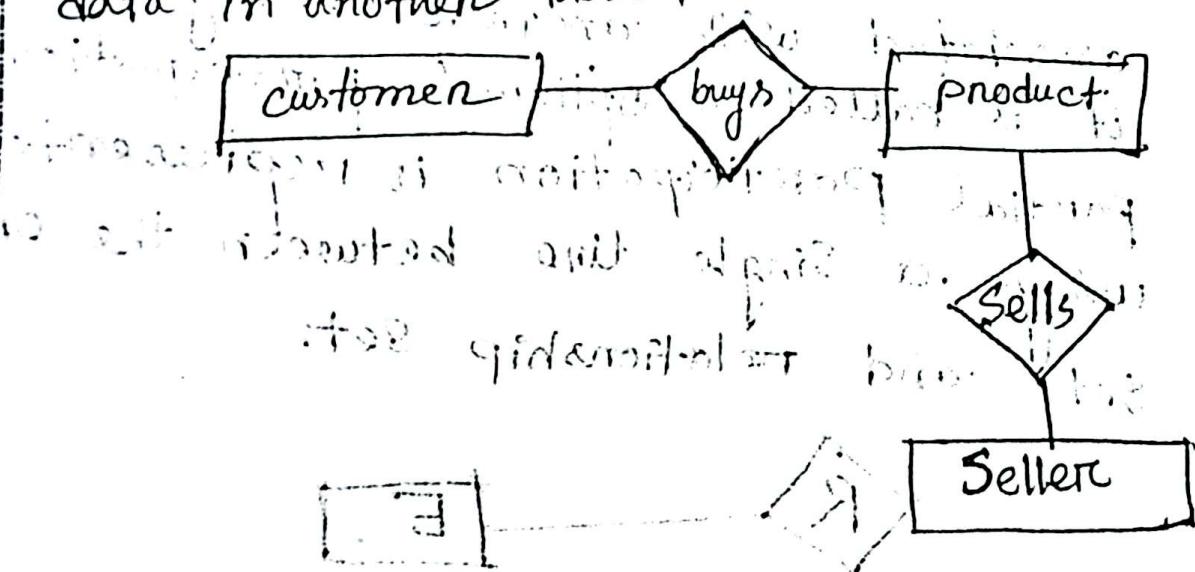


There's a single line between the entity set "course" and relationship set "Enrolled in" significant partial participation. It specifies there might exist some course for which no enrolment are made.

⑥ Define Relationship and Relationship set.

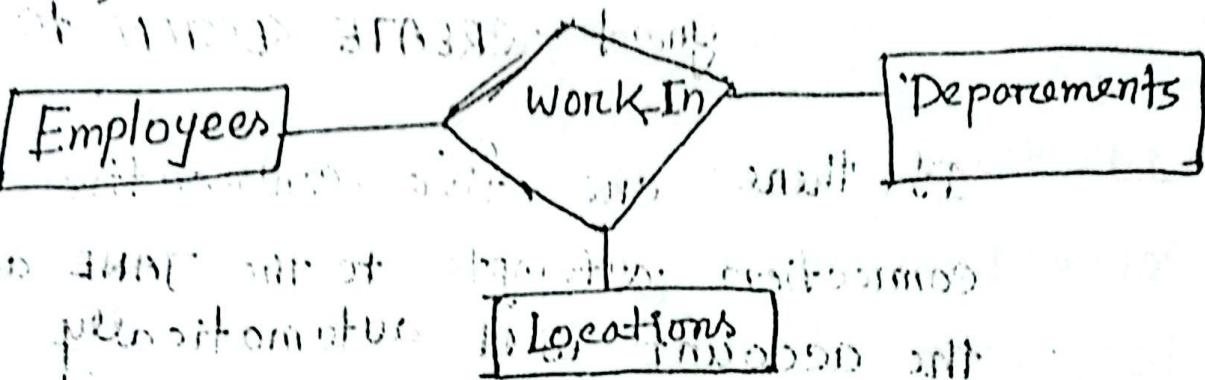
Relationship:

A relationship is a connection between two or more tables in a database. It refers to how the data in one table relates to the data in another table.



Relationship Set:

A relationship set is a collection of relationships between tables. It defines the connections between different sets of related data and specifies how they are linked.



Q. If you want to lock after 5 consecutive failed connection attempts to JANE account, what will have to do in Oracle? How?

I want to lock after five consecutive failed connection. Then I do in Oracle:

1. Open command prompt
2. Sqlplus
3. Username: System
4. Password: 1234

5. Create profile LIMITED_PROFILE limit
FAILED_LOGIN_ATTEMPT 5;
Create user JANE identified by EYRE
profile LIMITED_PROFILE;

grant CREATE SESSION to JANE;

If there are five consecutive failed connection attempts to the JANE account, the account will automatically locked by oracle. When you use the correct password for the JANE account, you will receive an error.

Connect Jane/Eyre

ERROR!
ORA-28000: the account is locked.

sqlplus .S

username : JANE password : EYRE

username : JANE password : EYRE

University wants to give scholarship to some students, on the following criteria:

- i. Students must be female.
- ii. Students do not get any other private scholarships such like DBBL Scholarship.
- iii. Grade must be at least 3.50
- iv. Student should not be punished for any awful activity.

Create necessary table and write necessary query.

Based on given criteria, we can create a table named Students with the columns Student-id, name, gender, has-private-Scholarship, is-punished.

Create table Students (Student-id VARCHAR(32) NOT NULL,
gender VARCHAR(10), name
grade FLOAT,
has-private-Scholarship BOOLEAN,
is-punished BOOLEAN);

University should meet the following conditions:

(i) Select * from Students where

gender = 'Female';

(ii) Select * from Students where

has_private_scholarship = FALSE;

(iii) Select * from Students where grade >= 3.50

(iv) Select * from Students where

is_punished = FALSE;

To give scholarship to some student on the following every criteria we can use this query:

SELECT * FROM Students

Where Gender = 'Female'

AND has_private_scholarship = FALSE

AND grade >= 3.50

AND is_punished = FALSE;

This query selects all female students who do not have private scholarship, have a CGPA of at least 3.50 and have not been punished for any awful activity.

Define weak entity set. Describe weak entity set with a suitable example.

weak entity set:

An entity set may not have sufficient attributes to form a primary key. Such entity set is called weak entity set.

The weak entity set which do not have sufficient attributes to form a primary key are known as weak entity. As the weak entity do not have any primary key, they can not be identified on their own. So, they need to depend on their other identity set, called strong identity set.

Instead of having primary key, weak entity set has partial key. Partial keys are set of attributes which helps the weak entity to be identified. The relationship associating the weak entity set with the identifying entity set is called the identifying relationship.

弱实体

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date _____

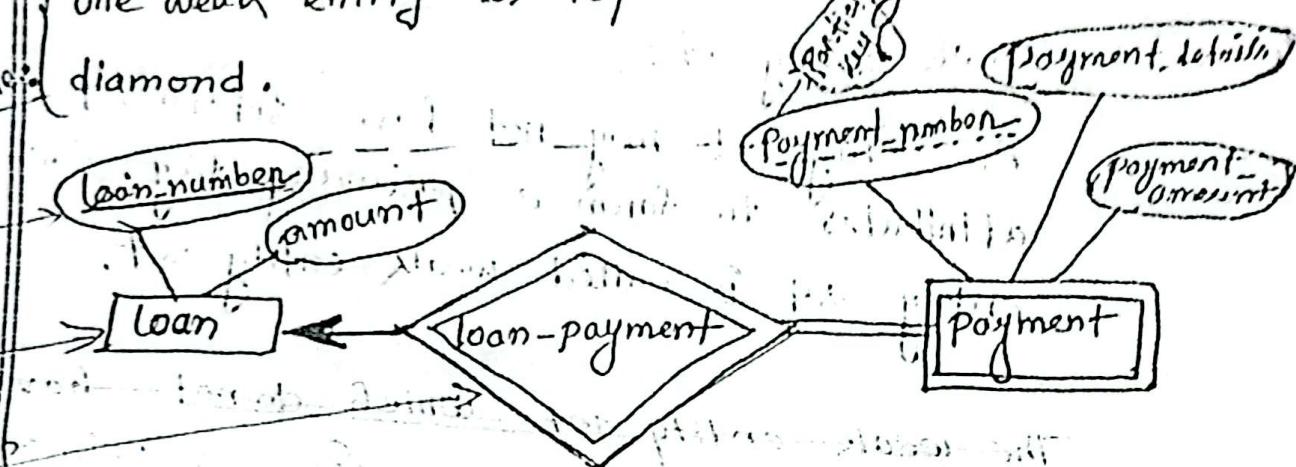
Weak entity set represented by double rectangle :: Relation between one strong and one weak entity is represented by double diamond.

Example

attribute

entity set

relationship
set
inc → link



loan-number এবং payment-number
কে আইডি করা যায়। তবু payment-number
কে আইডি করা যায় না। এবং payment-number
weak entity set। payment রিক আইডি
আবেদন করা প্রয়োজন এখ। এবং মুক্ত
identifying entity set / owner entity set এবং
এবং payment এর existent dependent এবং।

One-to-many relation → এবং loan এবং payment
কে আইডি করা যাবে। এবং মুক্ত

Total participation →

primary key হবে (loan-number, payment-number)

- ⑩ Describe each line of the following Trigger in Oracle.

Create or replace trigger : BOOKSHELF-BEF-UPD-ROW
 In this line trigger is named. The name of the trigger is "BOOKSHELF-BEF-UPD-ROW"?
 before update on BOOKSHELF

This trigger applies to the "BOOKSHELF" table.
 It will be executed before update transaction have been committed to the database.

For, "for each row" the trigger will apply to each row changed by the update statement.
 When ($\text{new.Rating} < \text{old.Rating}$)

When clause adds further criteria to the triggering condition. The triggering event not only must be an update of the bookshelf table, but also must reflect a lowering of the Rating value.

begin

insert into BOOKSHELF_AUDIT,

(Title, publisher, CategoryName,

:Old_Rating, New_Rating, Audit_Date)

old values

(:Old.Title, :Old.Publisher, :Old.CategoryName,

:Old.Rating, :New.Rating, Syndate);

represent Syndate inserted between old and new

The PL/SQL code shown in the following listing

is the trigger body. This example inserts

the old values from the BOOKSHELF record

into the BOOKSHELF_AUDIT table before

the BOOKSHELF record is updated.

is stoppage record from books table

Q) What is DB Trigger? What are the types of Triggers?

Trigger:

(A trigger is a special kind of stored procedure that is automatically executed in response to certain events or changes occurring within the database.)

The main purpose of using triggers is to enforcing business rules, auditing changes to data, and maintaining complex relationships between tables.

There are ~~too~~ many types of triggers:

1. Row level triggers:

This triggers fire once for each row affected by an INSERT, UPDATE or DELETE Statement. They can be used to enforce various types of data validation such as checking for duplicates or enforcing referential integrity constraints.

2. Statement-level-triggers:

These triggers fire once for each group of rows affected by an

INSERT, UPDATE OR DELETE statement.

They can be used to perform various types of processing or automation

tasks, such as logging changes or updating related records.

3. Before and After trigger

4. Instead of trigger

5. Schema trigger

6. Database-level triggers

• ETAGIO, TRIGGER are per table
• AT-based with transaction

• reatability able to support condition

• can be synchronous with payloads up to 1000 bytes

• synchronous propagation information

data automatically backup এবং delete/Edit
বাস্তু পর্যন্ত

Create or Replace trigger : t1.

before delete on main

for each row

begin

insert into Backup values (:old.id, :old.Salary);

end

► A trigger is a special kind of stored procedure that is automatically executed in response to certain events or changes occurring within the database.

Events :

1. Database Manipulation (DML): Delete, Insert or update
2. Database Definition (DDL): CREATE, ALTER OR DROP
3. Database operation → LOGON, LOGOFF, STARTUP OR SHUTDOWN

Benefits of triggers:

1. Gain strong control over security
2. Enforcing referential integrity
3. Event logging and storing information on table access
4. Auditing
5. Synchronous replication of tables.
6. preventing invalid transactions.
7. automatic column creation

Syntax:

CREATE TRIGGER trigger_name
ON table_name
FOR insert | update | delete
AS
BEGIN
[statements]
END

Example : (1) maintaining consistency of data

CREATE TRIGGER

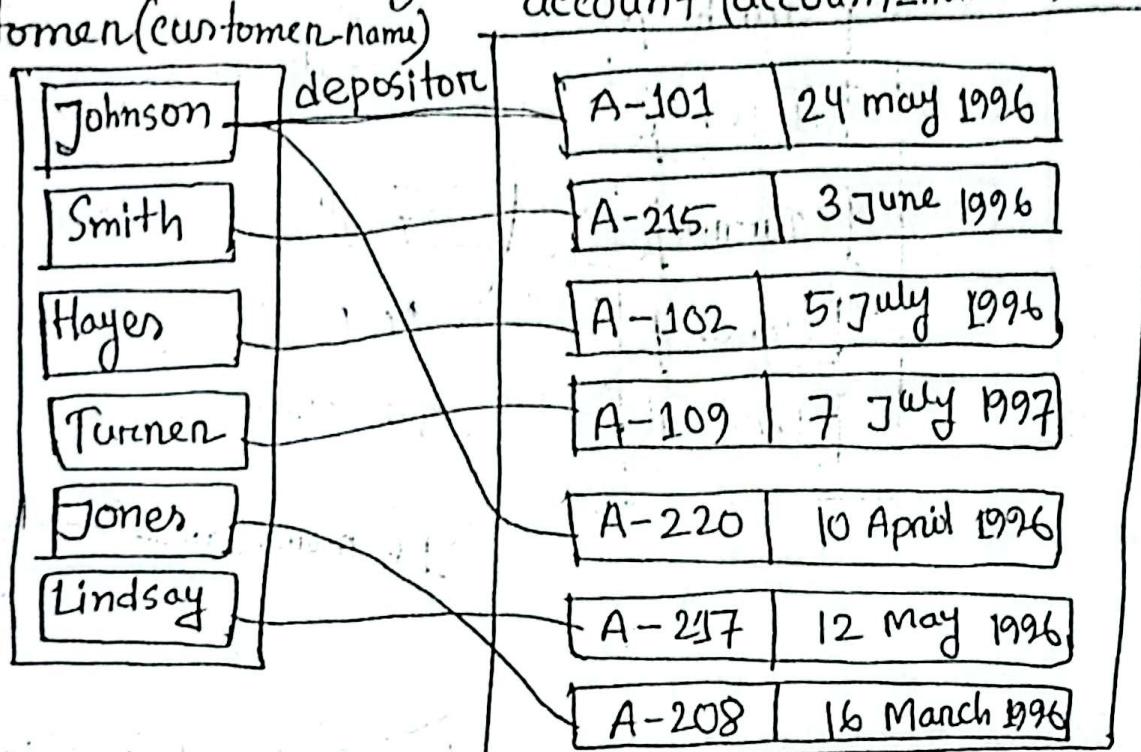
Example : (2) maintaining consistency of data

CREATE TRIGGER

Example : (3) maintaining consistency of data

12. The cardinality ratio of a relationship can affect the placement of relationship attributes - justify the above statement.

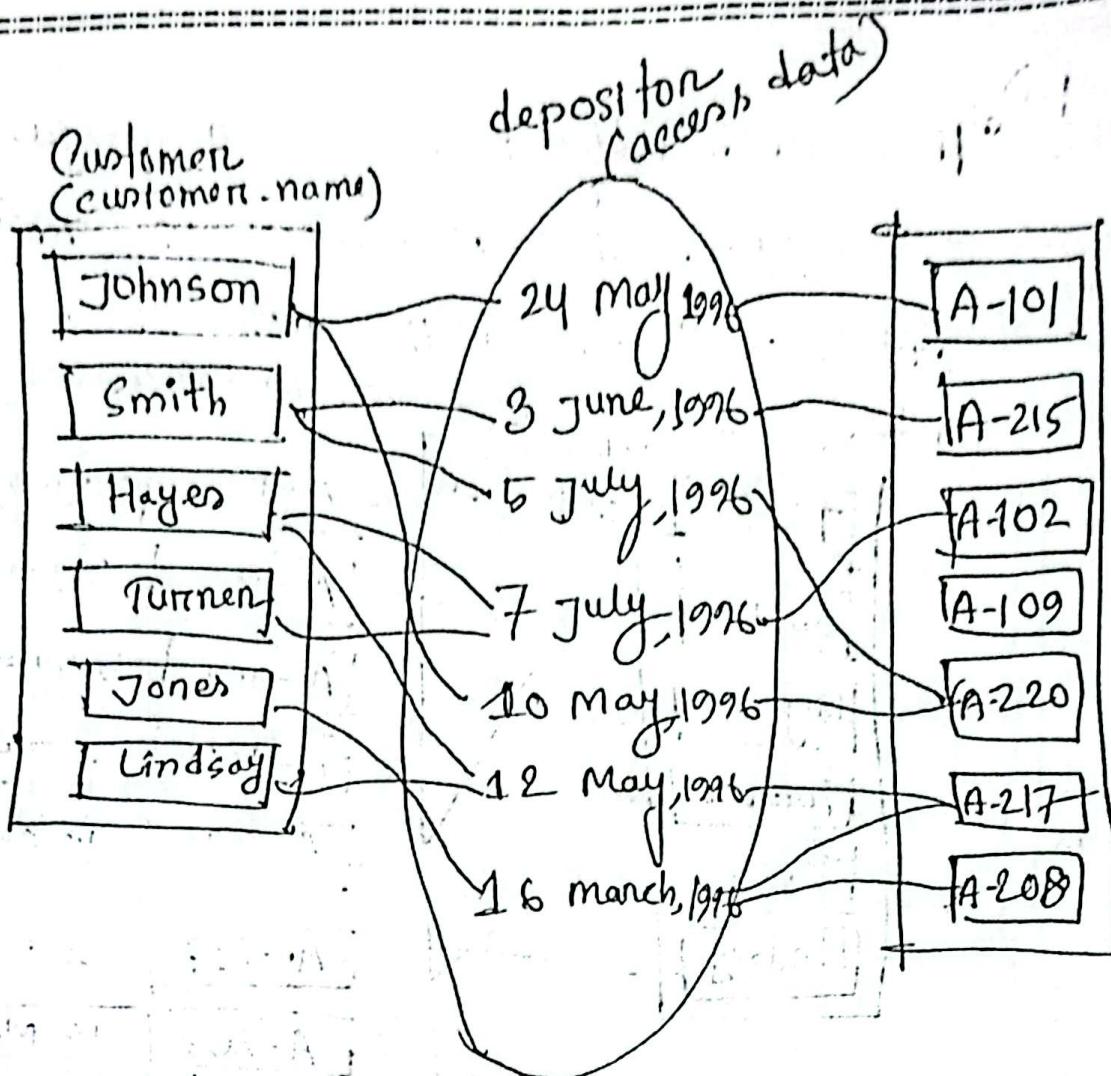
Customer (customer-name) account (account-number, access-data)



Access data as attribute of the account entity set

Fig: 01

Attributes of one-to-one or one-to-many relationship sets can be associated with one of the participating entity sets, rather than with the relationship set.



Access data as attribute of the depositor

relationship set : Customer

Fig-02

many to many relationship type some attributes can be determined by a combination of participating entities.

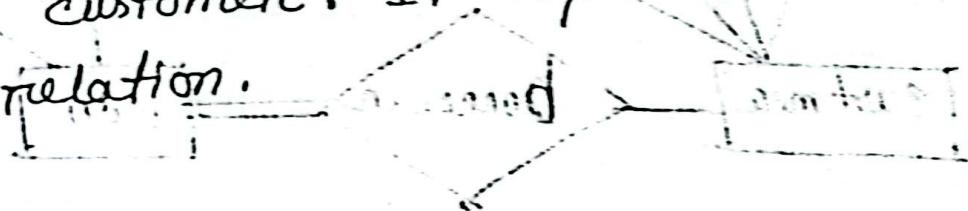
Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date: _____

In Fig-02 one customer have more than one account and one account can be access by more than one customer.

So, it is a many to many relation. In this case the time attribute stay at the middle of customer and account.

In Fig-01 one customer have one or more than one account but one account has one customer. It represent the one to many relation.

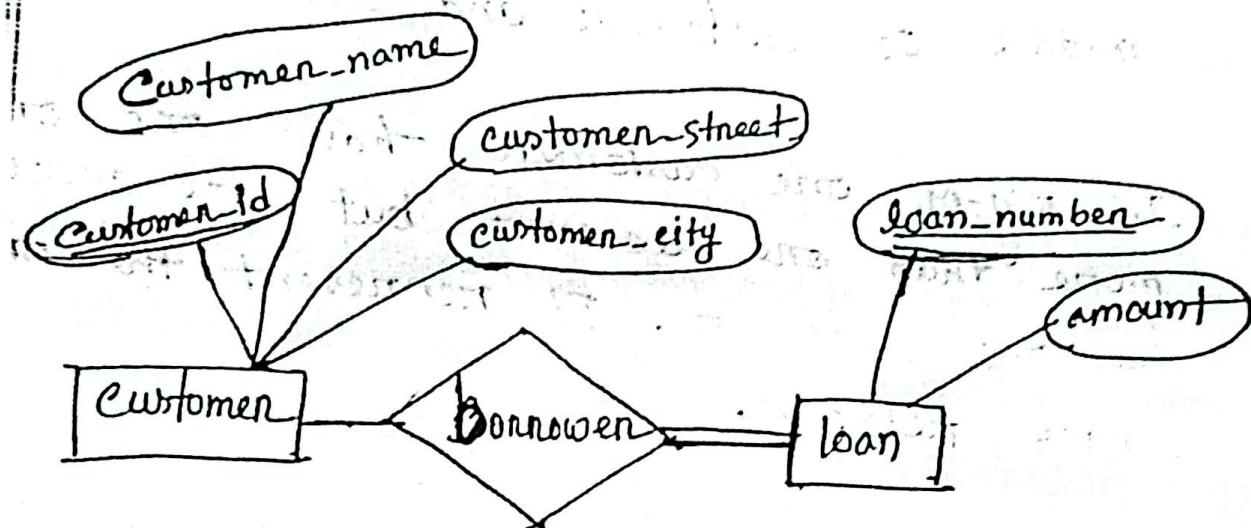


For this reason, we can say that the cardinality ratio of a relationship can affect the placement of relationship attribute. Now moving forward page 4!

multiple regions are used right side
page 4 and video capture wrong side

(x3) Every loan entity to be related to at least one customer through the borrower relationship;

In contrast, an individual can be a bank customer whether or not she has a loan with the bank. — Explain the statement clearly.



Loan Total-Participation

→ কোটি না কোটি যেহেতু loan নিবে

Customer partial participation

→ কোনো customer loan নিলেও সেই customer আছে।

* 5 নং প্রশ্নের উত্তর ও উপর্যুক্ত diagram

ব্যবসায় মাধ্যমে জটার উত্তর হুবে।

(ii) "password_REUSE_MAX and password_REUSE_TIME are mutually exclusive." - Is it true or false? Explain your answer.

These two parameters are mutually exclusive, to prevent a password from being reused, we use these parameters.

If you set a value for one of them, the other must be set to UNLIMITED.

The password_REUSE_Time parameter specifies the number of days that must pass before a password can be reused. For example, if you set `PASSWORD_REUSE_TIME` to 60, you cannot reuse the same password within 60 days.

The password_REUSE_Max parameter specifies the number of password changes that must occur before a password can be reused. If you attempt to reuse the password before the limit is reached, Oracle will reject your password change.

Sat / Sun / Mon / Tue / ...

Date: / /

For example, you can set `PASSWORD_REUSE_MAX` for the `LIMITED_PROFILE`, ~~created~~,

alter profile `LIMITED_PROFILE` limit
`PASSWORD_REUSE_MAX 3`

`PASSWORD_REUSE_TIME UNLIMITED`

If the user now attempts to reuse a recent password, the password change attempt will fail.

• If a user has already selected all previous password then it is rejected.
• If a user enters the same password again then it is rejected.

• If a user has entered his/her password for the first time then it is accepted.
• If a user has entered his/her password for the second time then it is accepted.
• If a user has entered his/her password for the third time then it is accepted.
• If a user has entered his/her password for the fourth time then it is accepted.
• If a user has entered his/her password for the fifth time then it is accepted.
• If a user has entered his/her password for the sixth time then it is accepted.
• If a user has entered his/her password for the seventh time then it is accepted.
• If a user has entered his/her password for the eighth time then it is accepted.
• If a user has entered his/her password for the ninth time then it is accepted.
• If a user has entered his/her password for the tenth time then it is accepted.

Sat / Sun / Mon / Tue / Wed / Thu / Fri
□ □ □ □ □ □ □

Date: _____

- (15) Create a trigger that is executed whenever an insert or an update occurs. What tables you need to create the trigger?

Create or replace trigger BOOKSHELF_BFE_UPD_INS_ROW
before insert or update of Rating on BOOKSHELF
for each row

begin

if inserting then

insert into BOOKSHELF_AUDIT

(Title, publisher, category-name,
New-rating, Audit-Date)

values

(:new.Title, :new.publisher, :new.category-name,
:new.Rating, sysdate);

else -- if not inserting then we are updating the Rating

Insert into BOOKSHELF_AUDIT

(Title, publisher, Category-Name, Old-Rating, New-Rating,
Audit-Date)

values

(:old.Title, :old.publisher, :old.Category-Name,
:Old.Rating, :new.Rating, Sysdate);

end if;

end;

Sat / Sun / Mon / Tue / Wed / Thu

Date / /

To implement this trigger I need two tables:

1. Bookshelf: This table already exist in my database.

2. BOOKSHELF-AUDIT: This table need to be created. It will used to store the audit information related to the changes in the Rating column.

The column should have columns such as Title, publisher, category_name, old_Rating, New_Rating and Audit_Date.

```
CREATE TABLE BOOKSHELF-AUDIT(
    Title VARCHAR(255),
    publisher VARCHAR(255),
    category_name VARCHAR(255),
    Old_Rating number,
    New_Rating number,
    Audit_Date DATE);
```

Q16 How one can enable and disable Triggers? 150

- By default, a trigger is enabled when it is created. However, there are situations in which you may want to disable a trigger.

Enable:

- To enable a trigger, use the alter trigger command with the enable keyword.

Ex:

alter trigger BOOKSHELF_BEF_UPD_INS_Row enable;

- To enabling all triggers we use the alter table command.

Ex:

alter table BOOKSHELF enable all triggers;

Disable:

- To disable a trigger, use the alter trigger command with the disable keyword.

Ex:

alter trigger BOOKSHELF_BEF_UPD_INS_Row disable;

- To disabling all triggers we use the alter table command.

Ex:

alter table BOOKSHELF disable all triggers;

564 → (17) Show an example where a procedure is called within triggers.

Rather than creating a large block of code within a trigger body, you can save the code as a stored procedure and call the procedure from within the trigger, by using call command.

Create or replace trigger BOOKSHELF_AFT_INS_ROW
after insertion on BOOKSHELF_AUDIT

for each row

begin

call INSERT_BOOKSHELF-AUDIT DUP (:new.Title,
:new.Publisher, :new.CategoryName, :new.OH_Rating,
:new.New_Rating, :new.AuditDate);

end;

which will allow → Trigger is design

to automatically trigger response to insert operations on the
books with also accept the primaryBOOKSHELF-AUDIT
constraint table.

To implement this identity constraint and trigger

Sat / Sun / Mon / Tue / Wed / Thu / Fri

Date : / /

Circulate a trigger BEFORE DELETE
triggers on the BOOKSHELF table. Where
conditions are:

When a user attempts to delete a record from the table, the trigger is executed and checks two system conditions:
that the day of the week is neither Friday
nor Saturday, and the Oracle username
of the account performing the delete
begins with the letters "CSE".

→ 233 pages

562 Page

11 hrs

soft copy

soft copy

University

• University Schema Diagram (293 page)

ER Diagram (293 page)

ANSWER

ANSWER

use

Customer Table Condition

Sat / Sun / Mon / Tue / Wed / Thu / Fri
□ □ □ □ □ □ □

Date / /

Create or replace trigger BOOKSHELF_BEF_DEL
before delete on BOOKSHELF
declare

: weekend_error exception;
not_CSE_user EXCEPTION;
begin

if TO_CHAR (SysDate, 'DY') = 'SAT' or
TO_CHAR (SysDate, 'DY') = 'FRI' Then
end if; Raise weekend_error;
if SUBSTR (User, 1, 3) <> 'CSE' Then
RAISE not_CSE_User;
end if;

Exception

When weekend_error Then
RAISE_APPLICATION_ERROR (-20001,
'Deletions not allowed on weekends');

When no_CSE_User

RAISE_APPLICATION_ERROR (-120002,
'Deletions only allowed by CSE users');

end;