VPM CLASSES

NTA UGC NET

COMPUTER SCIENCE & APPLICATIONS

SAMPLE THEORY - *(English Medium)*

* RELATIONAL DATABASE DESIGN
* NORMALIZATION

9001894070

www.vpmclasses.com

UGC NET - COMPUTER SCIENCE

SAMPLE THEORY

RELATIONAL DATABASE DESIGN

NORMALIZATION

VPM CLASSES

FOR IIT-JAM, JNU, GATE, NET, NIMCET AND OTHER Entrance Exams

Web Site www.vpmclasses.com E-mail - info@vpmclasses.com

RELATIONAL DATABASE DESIGN

Basic Concepts

- A database is a collection of logically related *records*
- A relational database stores its data in 2-dimensional *tables*
- A table is a two-dimensional structure made up of *rows* (*tuples*, *records*) and *columns (attributes*, *fields*)
- Example: a table of students engaged in sports activities, where a student is allowed to participate in at most one activity

Student	Activity	Fee
100	Cricket	200
150	Hockey	50
175	Football	50
200	Hockey	50

Table Characteristics

- Each row is unique and stores data about one entity
- Each column has a unique attribute name
- Each column (attribute) description (*metadata*) is stored in the database
- Access metadata is stored and manipulated via the Table Design View grid
- Row and column order is unimportant
- All entries in a column have the same data type
- Data type examples: Text(50), Number(Integer), Date/Time
- Each cell contains atomic data: no lists or sub-tables

Primary Keys

- A primary key is an attribute or a collection of attributes whose value(s) uniquely identify each row in a Relation
- A primary key should be minimal: it should not contain unnecessary attributes

Student	Activity	Fee
100	Cricket	200
150	Hockey	50
175	Football	50
200	Hockey	50

- We assume that a student is allowed to participate in at most one activity
- The only possible primary key in the above table is StudentID
- Sometimes there is more than one possible choice; each possible choice is called a candidate key
- What if we allow the students to participate in m ore than one activity?

Student	Activity	Fee
100	Cricket	200
100	Golf	65
175	Football	50
175	Hockey	50
200	Hockey	50
200	Golf	65

• Now the only possible primary key is the combined value of (StudentID, Activity),

• Such a multi-attribute primary key is called a composite key or concatenated key

Composite Keys

• A table can only have one primary key

• But sometimes the primary key can be made up of several fields

• Concatenation means putting two things next to one another: the concatenation of “burger” and “foo” is “burgerfoo”.

• consider the following table of cars

License Plate	State	Make	Model	Year
LVR120	RJ	Maruti	Swift	2003
BCX50P	NJ	Buick	Regal	1998
LVR120	CT	Toyota	Corolla	2002
908HYY	MA	Ford	Windstar	2001
UHP33	NJ	Tata	Nano	2006

• LicensePlate is not a possible primary key, because two different cars can have the same license plate number if they’re from different states

• But if we concatenate LicensePlate and State, the resulting value of (LicensePlate, State) must be unique:

• Example: “LVR120NJ” and “LVR120CT”

• Therefore, (LicensePlate, State) is a possible primary key (a candidate key)

• Sometimes we may invent a new attribute to serve as a primary key (sometimes called a synthetic key)

• If no suitable primary key is available

• Or, to avoid composite keys

• In Access, "Autonumber" fields can serve this Purpose

Foreign Keys

• A foreign key is an attribute or a collection of attributes whose value are intended to match the primary key of

Some related record (usually in a different table)

• Example: the STATE and CITY table below

State Table:

State Abbrev	State Name	Union Order	State Bird	State Population
RJ	Rajasthan	5	Peacock	3,287, 116
UP	Uttar Pradesh	26	sparrow	9,295, 297
WB	West Bengal	40	whit e Pigeon	696,004
MP	Madhya Pradesh	16	Hen	4, 877.185
AS	Assam	28	Crow	16, 986, 510

City Table :-

State Abbrev	City Name	City Population
RJ	Kota	139, 739
RJ	Jaipur	14,031
RJ	Jodhpur	8,418
UP	Mathura	127,321
W B	Trip ura	6,257
WB	Patna	12,906
MP	Bhopal	488,374
AS	Nagaland	465,622
AS	manipur	12,224

• Primary key in STATE relation **: StateAbbrev**

• Primary key in CITY relation: **(StateAbbrev, CityName)**

• Foreign key in CITY relation: **StateAbbrev**

Database Anomalies

• Anomalies are problems caused by bad database design example: ACTIVITY(StudentID, Activity, Fee)

Student	Activity	Fee
100	Cricket	200
100	Golf	65
175	Foo tball	50
175	Hockey	50
200	Hockey	50
200	Golf	65

• An insertion anomaly occurs when a row cannot be added to a relation, because not all data are available (or one has to invent "dummy" data)

Example: we want to store that volleyball costs $175, but have no place to put this information until a student takes up volleyball (unless we create a fake student)

• A deletion anomaly occurs when data is deleted from a relation, and other critical data are unintentionally lost

Example: if we delete the record with StudentID = 100, we forget that cricket costs $200

• An update anomaly occurs when one must make many changes to reflect the modification of a single datum

Example: if the cost of Hockey changes, then all entries with Hockey Activity must be changed too

Cause of Anomalies

• Anomalies are primarily caused by:

(1) *Data redundancy*: replication of the same field in multiple tables, other than foreign keys

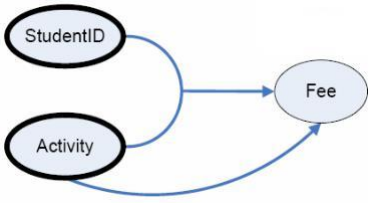
(2) Functional dependencies whose determinants are not candidate keys, including

Partial dependency

Transitive dependency

• Example: ACTIVITY (StudentID, Activity, Fee)

Student	Activity	Fee
100	Cricket	200
100	Golf	65
175	Foo tball	50
175	Hockey	50
200	Hockey	50
200	Golf	65



• Activity by itself is not a candidate key, so we get anomalies (in this case, from a partial dependency)

Fixing Anomalies (Normalizing)

• Break up tables so all dependencies are from prim ary (or candidate) keys

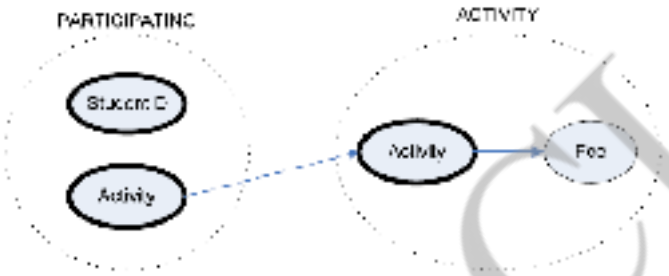
PARTICIPATING (StudentID, Activity)

Activity foreign key to ACTIVITIES

ACTIVITY(Activity, Fee)

Stud entID	Activity
100	Cri cket
100	Golf
150	Hockey
175	Football
175	Hockey
200	Hockey
200	Golf

Activity	Fee
Cricket	200
Golf	65
Hockey	50
Football	50
Volleyball	200



StudentID	Acti vity
100	Cricket
100	Golf
150	Hockey
175	Footba ll
175	Hockey
200	Hockey
200	Golf

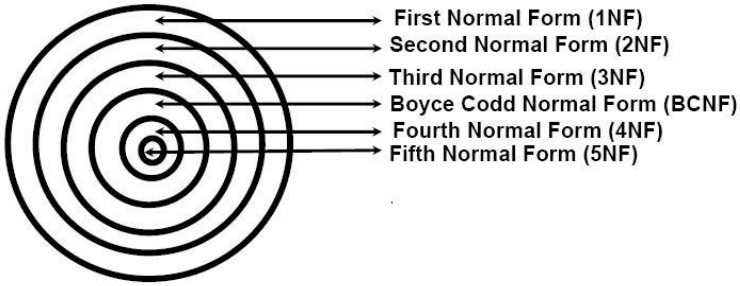
Activity	Fee
Cricket	200
Golf	65
Hockey	50
Football	50
Volleyball	200

• The above relations do not have any of the anomalies

NORMALIZATION

• “Normal forms” are categories that classify how completely a table has been normalized

• There are six recognized *normal forms* (*NF*):



Functional Dependency

(1) Functional Dependency Is a relationship between or among attributes such that the values of one attribute depend on, or are determined by, the values of the other attribute(s).

(2) Partial Dependency: Is a relationship between attributes such that the values of one attribute is dependent on, or determined by the values of another attribute which is part of the composite key. Partial Dependencies are not good due to duplication of data and update anomalies.

Examples of Functional Dependencies:

- If we know an ISBN, then we know the Book Title and the Author(s)

  (1)ISBN Book Title

  (2)ISBN Author(s)

- If we know the VIN, then we know who is the Auto owner

  VIN Auto_Owner

- If we know Student-ID (SID), then we can uniquely determine his/her Name

  SID S_Name

Transitive Dependencies

- Is a relationship between attributes such that the values of one attribute is dependent on, or determined by, the values of another attribute which is not a part of the key.
- Exist when a nonkey attribute value is functionally dependent upon another nonkey value in the record.

For Example:

EMPLOYEE_ID --> JOB_CATEGORY

JOB_CATEGORY --> HOURLY_RATE

- An employee data table that includes the “hourly pay rate” would require searching every employee record to properly update an hourly rate for a particular job category.

GOLDEN RULE OF NORMALIZATION: Enter the minimum data necessary, avoiding duplicate entry of information with minimum risks to data integrity.

Goals of Normalization:

(1) Eliminate redundancies caused by:

(a) Fields repeated within a file

(b) Fields not directly describing the key entity

(c) Fields derived from other fields

(2) Avoid anomalies in Updating (Adding, Editing, Deleting)

(3) Represent accurately the items being modeled

(4) Simplify maintenance and retrieval of info

Database Tables and Normalization

- Normalization is a process for assigning attributes to entities. It reduces data redundancies and helps eliminate the data anomalies.
- Normalization works through a series of stages called normal forms:

(1) First normal form (1NF)

(2) Second normal form (2NF)

(3) Third normal form (3NF)

- The highest level of normalization is not always desirable.

Basic Rule for Normalization

- The attribute values in a relational table should be functionally dependent (FD) on the primary key value.

(1) A relationship is functionally dependent when one attribute value implies or determines the attribute value for the other attribute.

EM_SS_NUM --> EM_NAME

- Corollaries

(1) Corollary 1: No repeating groups allowed in relational tables.

(2)Corollary 2: A relational table should not have attributes involved in a transitive dependency relationship with the primary key.

Normalization Benefits

- Facilitates data integration.
- Reduces data redundancy.
- Provides a robust architecture for retrieving and maintaining data.
- Compliments data modeling.

- Reduces the chances of data anomalies occurring.

The data redundancies yield the following anomalies:

(a) Update anomalies.

(b) Addition anomalies.

(c)) Deletion anomalies.

Deletion Anomaly

- Occurs when the removal of a record results in a loss of important information about an entity.
- **Example:**

  All the information about a customer is contained in an Order file, if the order is canceled, all the customer Information could be lost when the order record is deleted
- **Solution:**

  Create two tables--one table contains order information and the other table contains customer information.

Update Anomaly

- Occurs when a change of a single attribute in one record requires changes in multiple records
- **Example:**

  A staff person changes their telephone number and every potential customer that person ever worked with has to have the corrected number inserted.
- **Solution:**

  Put the employees telephone number in one location--as an attribute in the employee table.

Insertion Anomaly

- Occurs when there does not appear to be any reasonable place to assign attribute values to records in the database. Probably have overlooked a critical entity.
- **Example:**

  Adding new attributes or entire records when they are not needed. Where do you place information on new Evaluator's ? Do you create a dummy Lead.
- **Solution:**

  Create a new table with a primary key that contains the relevant or functional dependent attributes.
- **Conversion to First Normal Form**

  (a) A relational table must not contain repeating groups.

  (b) Repeating groups can be eliminated by adding the entry in at least the primary key column(s).

PROJ_ NUM	PROJ_NAME	EMP _ NUM	EMP_NUM	JOB_ CLASS	CHG HOUR	Hours
15	Greenfield	103	Mohan	Elect. Engineer	$84.50	23.8
		101	Ram	Database Designer	$105.00	19.4
		105	Amit	Database Designer	$105.00	35.7
		106	Shyam	Programmer	$35.75	12.6
		102	Puneet	Sy stems Analyst	$96.75	23.8

Database Table 1. The Evergreen Data

- Dependency Diagram

(a) The arrows above the entity indicate that the entity's are dependent on the combination of

PROJ_NUM and EMP_NUM .

(b)The arrows below the dependency diagram indicate less desirable dependencies based on only a part of the primary key -- partial dependencies.

A Dependency Diagram: First Normal Form

- **1NF Definition**

(1) The term first normal form (1NF) describes the tabular format in which:

(a) All the key attributes are defined.

(b)There are no repeating groups in the table.

(c)) All attributes are dependent on the primary key.

- Conversion to Second Normal Form

(1) Starting with the 1NF format, the database can be converted into the 2NF format by

(a) Writing each key component on a separate line, and then writing the original key on the last line

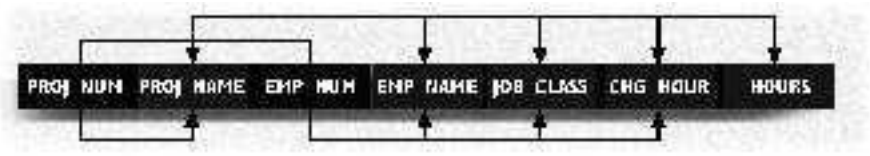
(b) Writing the dependent attributes after each new key.

PROJECT (PROJ_NUM, PROJ_NAME)

EMPLOYEE (EMP_NUM, EMP_NAME,

JOB_CLASS,CHG_HOUR)

ASSIGN (PROJ_NUM, EMP_NUM, HOURS)



- **2NF Definition**

(1) A table is in 2NF if:

(a) It is in 1NF and

(b) It includes no partial dependencies; that is, no attribute is dependent on only a portion of the primary key.

- **Note:**

It is still possible for a table in 2NF to exhibit transitive dependency; that is, one or more attributes may be functionally dependent on non key attributes.

- **Conversion to Third Normal Form**

Create a separate table with attributes in a transitive functional dependence relationship.

PROJECT (PROJ_NUM, PROJ_NAME) ASSIGN

(PROJ_NUM, EMP_NUM, HOURS) EMPLOYEE

(EMP_NUM, EMP_NAME, JOB_CLASS) JOB

(JOB_CLASS, CHG_HOUR)

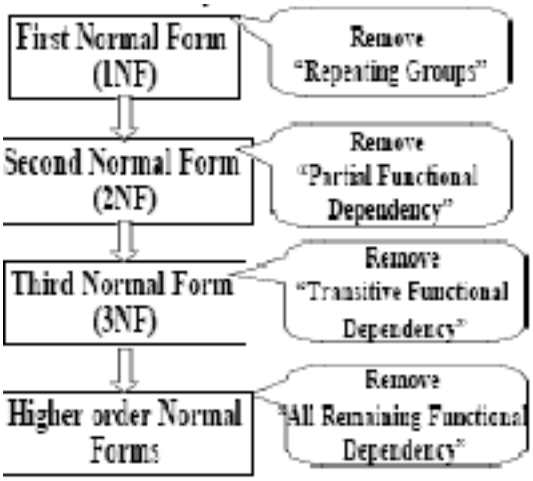
- **3NF Definition**

A table is in 3NF if:

(a) It is in 2NF and

(b) It contains no transitive dependencies.

Summary of one conversion form to another



SOLVED EXAMPLES

1. Every time attribute A appears, it is matched with the same value of attribute B, but not the same value of attribute C. Therefore, it is true that:

(A) A → B

(B) A → C

(C) A → (B,C).

(D) (B,C) → A.

2. In the relational model, relationships between relations or tables are created by using:

(A) Composite keys.

(B) Determinants.

(C) Candidate keys.

(D) Foreign keys

3. A tuple is a(n) :

(A) Column of a table.

(B) Two dimensional table.

(C) Row of a table

(D) Key of a table.

4. Which of the following is not a restriction for a table to be a relation?

(A) The cells of the table must contain a single value.

(B) All of the entries in any column must be of the same kind.

(C) The columns must be ordered

(D) No two rows in a table may be identical.

5. For some relations, changing the data can have undesirable consequences called:

(A) Referential integrity constraints.

(B) Modification anomalies

(C) Normal forms.

(D) Transitive dependencies.

Hints & Solutions

1. (A), **2.** (D), **3.**(C), **4.** (C), **5** (B)

1. **(A)** Because only value of A will match with B.
2. **(D)** Because a foreign key is an attribute or a collection of attributes whose value are intended to match the primary key of some related record (usually in a different table)
3. **(C)** A table is a two-dimensional structure made up of rows (tuples, records) and columns (attributes, fields)
4. **(C)** Column order is unimportant in Relational database design
5. **(B)** An update anomaly occurs when one must make many changes to reflect the modification of a single datum