DATABASE SYSTEM

What is database system

- Database system is a shared integrated computer structure that stores a collection of end user data and metadata.
- The data in the database are organized in the form of tables and stored in the form of data files in the database.
- End user data is a raw fact of interest of end user while metadata is the definition of end user data (data about data)

Different between data and information

- Data are raw facts, the word raw indicates that the facts have not yet processed to reveal meaning**.**
- Information is the results of processing raw data to reveal its meaning in short information is generated from processed data.
- Information is a key to good decision making.

Data quality and data governance

- The quality of the data within the database is essential if the organization is to make accurate short- and long-term business decisions.
- Data quality levels
  - ✓ Accuracy
  - ✓ Relevance
  - ✓ Completeness
  - ✓ Timeliness
  - ✓ Uniqueness
  - ✓ Unambiguous

Database management system

- DBMS is a collection of programs that manages the database structure and control access to the data stored in the database.
- DBMS enables the data in database to be shared among multiple applications or user.

ADVANTAGE OF DBMS

- Improve data sharing
  - ✓ The DBMS helps in creating environment in which end user get better access to more and better managed data.
- Better data integration
  - ✓ Wider access of well managed data promotes an integrated view of the organization's operation and clear view of big picture.
- Minimized data inconsistency

- ✓ Data inconsistency exists when different version of the same data appears in different places.

❖ Improve data access
- ✓ The DBMS makes possible to produce quick answer to ad hoc queries.
- ✓ Query is a specific request for data manipulation.

❖ Improve decision making
- ✓ Better managed data and improve data access make it possible to generate better quality information, on which better decision based.

❖ Increase end user productivity
- ✓ Availability of data, combined with the tools that transform data into useable information, empower end user to quick, informed decision that can be different between success and failure

Type of databases

❖ Database are classified according to:
- ✓ the number of users
  - determines whether the database is a single user or multiuser.
  - Where single user database support one user at a time. While multiuser database support multiple user at a time.
  - If multiuser database support less than 50 user or specific department in the organization, it is called workgroup database.
  - If multiuser database support more than 50, it is called enterprise database
- ✓ where the data located.
  - If database support data located in one site is called centralized database.
  - If database support data located in several sites is called distributed database
- ✓ the type of stored
- ✓ the intended data usage
  - a database that is designed to support a company's day to day operations is called operational database.
- ✓ degree to which the data are structured.

Data Redundancy

❖ data redundancy exists when the same data stored unnecessary at different places.

❖ Uncontrolled data redundancy course the following problems or stages:
- ✓ Data inconsistency
  - Data inconsistency exists when different version of the same data appears in different places in database system.
    Example done in class
- ✓ Poor data security
  - Having multiple copies of data increase the chances for a copy of data to be accessed by unauthorized user.
- ✓ Data anomalies

- Data anomaly develops when all required changes in the redundant data are not made successfully.
- Data anomalies includes:
    - Update data anomaly
        - ❖ Update data anomaly exist when trying to update a redundant data.
        - ❖ Deletion data anomaly exist when trying to delete a redundant data.
        - ❖ Insertion data anomaly exist when trying to insert redundant data.

The database system Environment.

- ❖ Database system refers to an organization of components that define and regulate the collection, storage, management and use of data within a database environment.
- ❖ Database system is composed of five components which are:
    - ✓ People
        - Includes all users of a database systems, are named based on primary job or function of the user.
        - Consist of five users namely:
            - System administrator
                - Oversee the database system general operations.
            - Database Administrator
                - Manages the DBMS and ensure that the database is functioning well.
            - Database designer
                - Design the database structure
                - If design is poor, even the performance of the database will be poor.
            - System analyst and programmers design and implement the application programs
                - They design and create the data screens, reports and procedure through which end user access and manipulate the database data.
            - End user
                - Are the people who use the application programs to run organization daily operations.
    - ✓ Hardware
        - ❖ Refers the physical components of the system for example storage devices, computer, network devices etc.
    - ✓ Software
        - ❖ Makes database system function fully, three type of software:
            - Operating software

- Manages all hardware components and makes it possible for all other software to run on computer.
- Examples Microsoft windows, Linux etc.

- Application software and utility
  - Used to access and manipulate data in the DBMS and to manage the computer environment in which data access and manipulation take place.
  - Example: MYSQL, oracle application and Microsoft access.
- DBMS software
  - Manages the database within the database system.
  - Example Microsoft access and SQL server, oracle corporation's oracle and IBM's DB2.

✓ Procedure
- ❖ Procedure are the instruction and rules that govern the database design and the use of database system.
- ❖ Good example of procedure, business rules.

✓ Data
- ❖ The word data covers the collection of facts stored in the database.

Question

A. Different between data dependence and independence.
B. List and discuss functions of DBMS.
C. Discuss why database design is important?
D. What are the potential costs of implementing a database system.
E. Discuss each of the following terms
   I. Field
   II. Data
   III. Record
   IV. File

Chapter 2: Data models

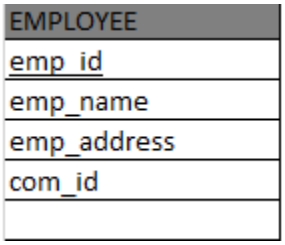
What is a meaning of data model.

- ❖ Data model is a representation of the complex real-world data structure in the form of a diagram (graphical representation of a real-world data structure)
- ❖ The main function of data model is to help to understand the complexities of the real-world environment.
- ❖ In database environment represents data structure and their characteristics, relationship, constraints and transformation.

Data models basic building blocks

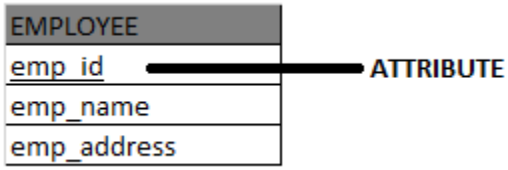
❖ Entity
- ✓ Is anything about which data are to be collected and stored.

- ✓ Also known as relation or table
- ✓ Entity in data model is represented by rectangle
- ✓ Example of entity



- ❖ Attribute
  - ✓ An attribute is a characteristic of an entity.
  - ✓ Also know as a column or tuple
  - ✓ Attributes on data model are represented by oval around entity object or below entity name.

Example



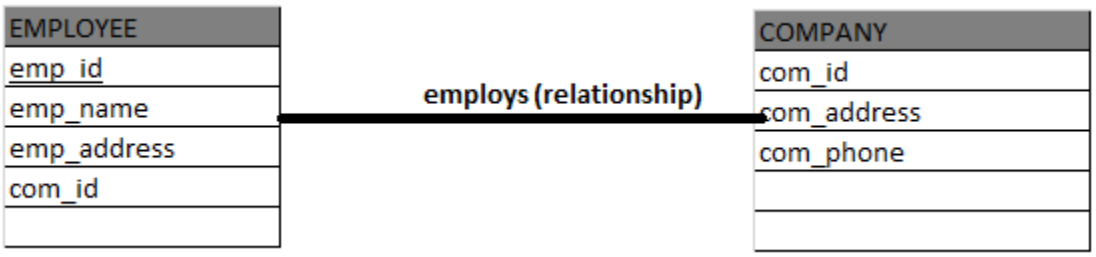
- ❖ Relationship
  - ✓ Relationship describes an association among entities
  - ✓ On the data model are represented by a diamond between entities or written on the line link two entities.
  - ✓ Relationship is represented by verb in the expression or in the business rule.

Example:

company employs many employees.

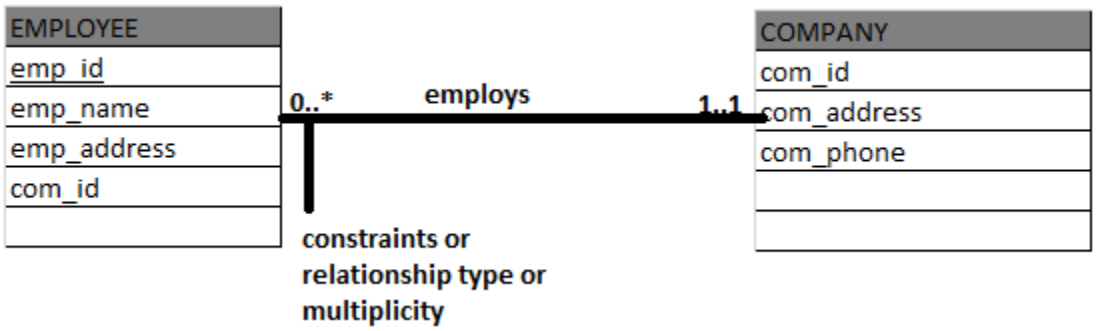
Employs is the relationship between company and employee.

Diagram representation:



- Constraints
  - ✓ A constraint is a restriction placed on the data.
  - ✓ Constraint helps to enforce data integrity.
  - ✓ Write next to entity box on the entity relationship diagram

Example



Type of relationship or constraints or multiplicity or cardinality.

I. **One to Many**
II. **Many to Many**
III. **One to One**
IV. **Many to One**

Business Rule

- A business rule is a brief, precise and unambiguous description of a policy, procedure or principle within a specific organization.
- Example of business rules are as follows
  - ✓ Company employs many employees on a contract based.
  - ✓ One employee employed by one company.

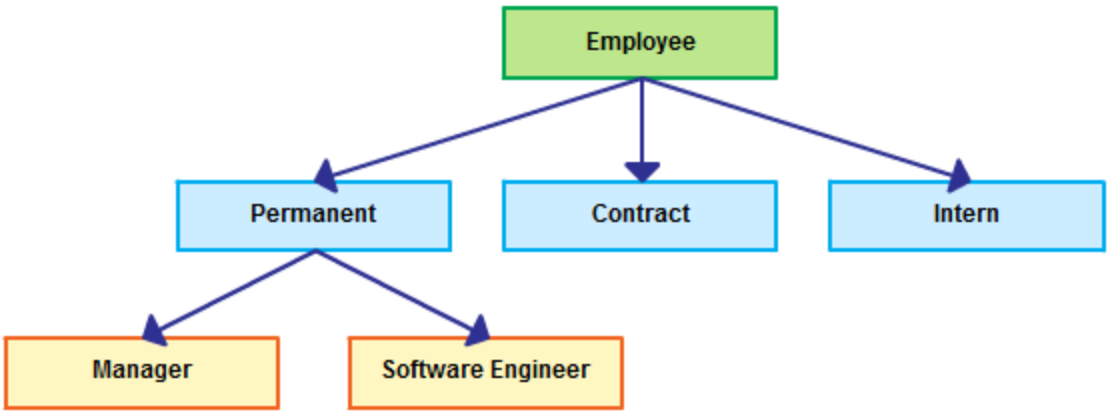
Translating business rules into data model components

N.B done in class.

Types of data models

- **Hierarchical Model**
  - ✓ The model's basic logical structure is represented by an upside-down tree.
  - ✓ The hierarchical structure contains levels or segments
  - ✓ A segment is equivalent to a file system record type.
  - ✓ The higher segment level known as the parent segment and below known as child segment.

Example:



- ❖ network models.
  - ✓ Network model was created to represent complex data relationships more effective than hierarchical model.
  - ✓ To improve database performance.
  - ✓ Network model allows a record to have more than one parent.

Example:

- ❖ The relational model
  - ✓ The relational model represented a major break-through for both users and designers.
  - ✓ The relational model uses RDBMS to manage tables in the database.
  - ✓ RDBMS performs the same functions as DBMS.
  - ✓ Each table is a matrix of row/columns intersections.

Example of relational model

Relational Model

Activity Code	Activity Name
23	Patching
24	Overlay
25	Crack Sealing

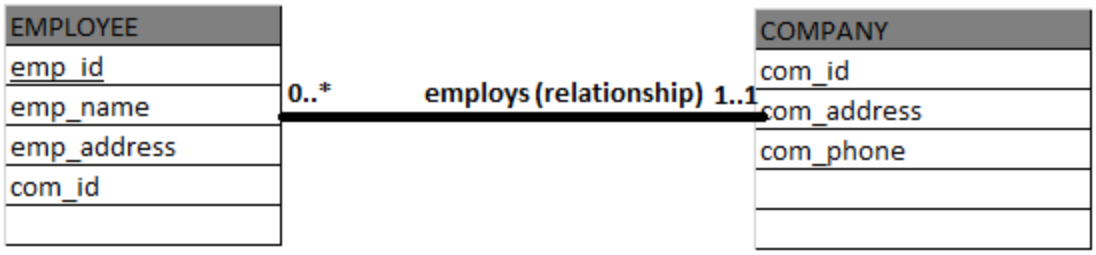
Key = 24

Activity Code	Date	Route No.
24	01/12/01	I-95
24	02/08/01	I-66

Date	Activity Code	Route No.
01/12/01	24	I-95
01/15/01	23	I-495
02/08/01	24	I-66

- The entity relationship model
  - ✓ ERM are normally represented in a entity relationship diagram ERD), which use graphical representation to model database component.
  - ✓ Entity relationship diagram are representation into three notations namely:
    - a) Unified modelling language (UML)
      - ✓ The different of the notation is only representation of the relationship type or connectivity.
      - ✓ UML use * for many and 1 for one.

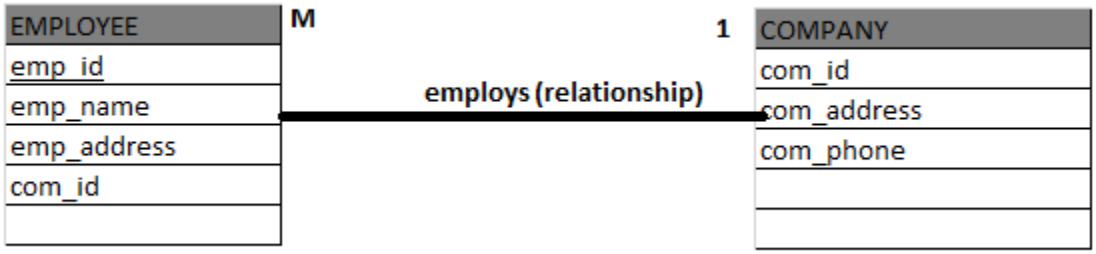
Example of ERD using UML notation:



b) Chen model notation

- ✓ CHEN's model use M or N for many and 1 for one written next to entity box.

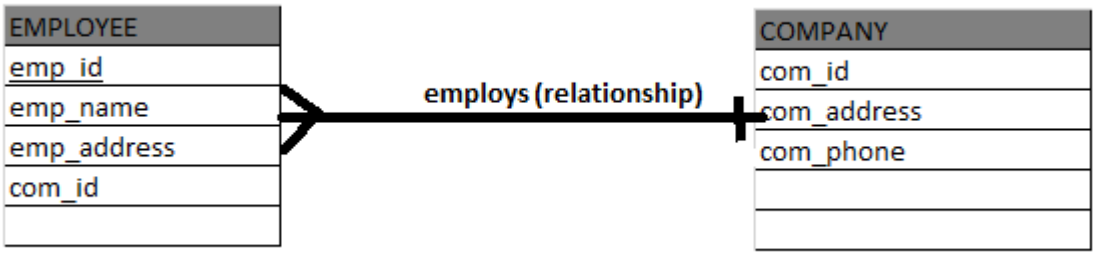
Example using chen's model notation



c) Crow's foot notation

- ✓ Crows foot notation uses less than and greater than symbol for many, the sign must be written between relationship link and attached to entity box
- ✓ Then one represented by one a cross relationship link.

Example using crow's foot notation



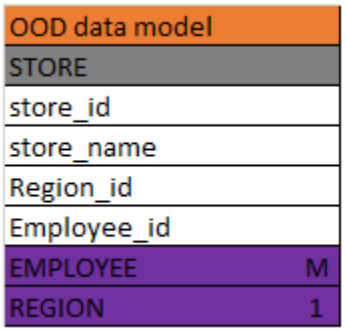
- ❖ Object oriented model
  - ✓ OOD model both data and relationship are contained in a single structure known as object.
  - ✓ Object oriented data model use OODBMS to manage data access and manipulation in the database.

- ✓ The object-oriented data model also known as sematic data model.

Components of OODM

- Object is an abstraction of a real-world entity.
- Attribute describes the properties of an object.
- Objects that share similar characteristic are grouped in a class.
- Inheritance is the ability of the object to inherit the attributes and the methods of the class above it.
- The OOD model represents objects as a box, contains all relationships and related objects attributes to other objects are included in one object box.
- The ER model uses three separated entities and two relationships to represent an invoice transaction.

Example of object-oriented data model.



DEGREEOF DATA ABSTRACTIONS

- ❖ Degree of data abstraction define the levels of data view in the database environment which includes following levels:
  - ✓ External model
    - The external model is the end user view of the data environment.
    - The end user refers to people who use the application programs to manipulate the data and generates information.
  - ✓ Conceptual model
    - Conceptual view is the database designer view.
    - Model use graphical representation of data in the form of models.
  - ✓ The internal model
    - The internal model is the DBMS view.
    - Data is represented in the form of internal schema
    - DBMS can only read the SQL statements using DDL and DML languages.
  - ✓ The physical model
    - The physical model operates at the lowest level of abstraction.
    - The physical model describes the way data are stored in the storage device.

QUESTIONS

1. Discuss the importance of data modelling.
2. Convert the following business rules into ERD using UML notation.
   A department employs many Employee, but each employee is employed by one department. Some employees, known as rover's are not assigned to any department. A division operates many departments, but each department is operated by one division. An employee may be assigned many projects, and a project may have many employees assigned to it. A project must have at least one employee assigned to it. One of the employees manages each department, and each department is managed by only one employee. One of the employees runs each division, and each division is run by only one employee.
3. Draw and discuss the stages of database life cycle.

ENTITY RELATIONSHIP MODELS

Entity

- Entity is any object we are interested in for data collection.
- Entity relationship model refers to a table row as the entity instance or occurrence.
- Entity consist of two types of entity which are:

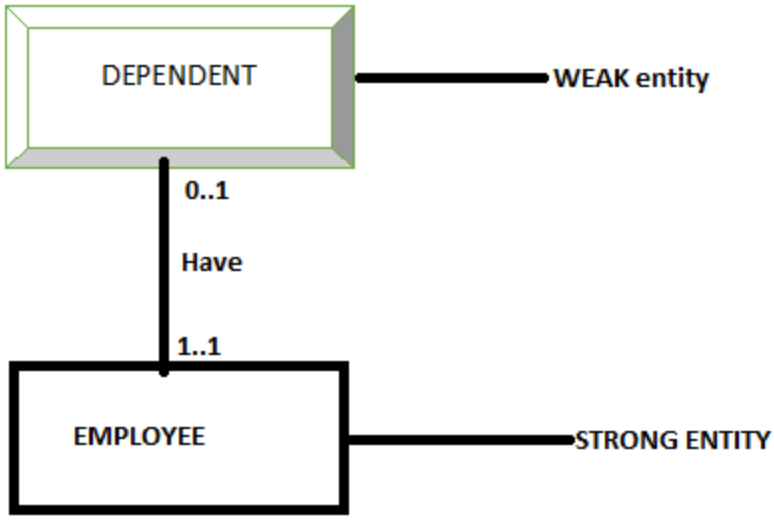
    - Weak entities
        - A weak entity is one that meets two conditions:
            1. The entity is existence dependent to the parent entity, cannot exist without entity which they are in the relationship.
            2. The entity primary key is partially or totally derived from the parent entity, normally the use of composite key defines a weak entity.
        - Weak entity is represented by double- walled rectangle.

Example give the following relational schema to determine the whether entity is weak or not

EMPLOYEE (EMP**_ID**, EMP_NAME, EMP_LNAME, EMP_DOB) AND DEPENDENT (**EMP_ID, DEP_ID**, DEP_NAME, DEP_DOB)

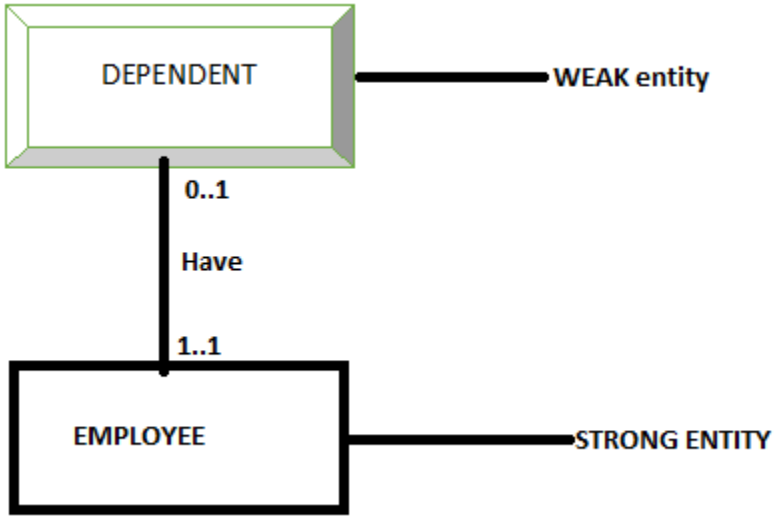
From the relational schema above we will determine that dependent table is weak entity because the primary key contains the primary key of the parent entity, in short that if employee table can delete from the database also dependent table will be delete therefore dependent table depends on the existence of employee table.

Diagram:



- Strong entities
  - ❖ The strong entity is the independent entity and does not use composite primary key to so that the entity is independent.
  - ❖ Representation of strong entity in the entity model, we use a rectangle.

Entity relationship diagram between EMPLOYEE and DEPENDENT tables



Example if we can consider relational schema above we can conclude that employee table is independent because it uses primary key EMP_ID not composite key.

- Composite entity
  - Composite entity is an entity that is use to break up M:M relationship into 1:M and M:1 relationship.

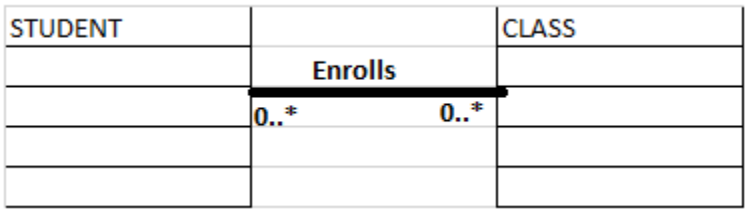
- Also known as bridge entity.

Example consider the business rule below:

Each student can enroll in several class while each class enrolls many students

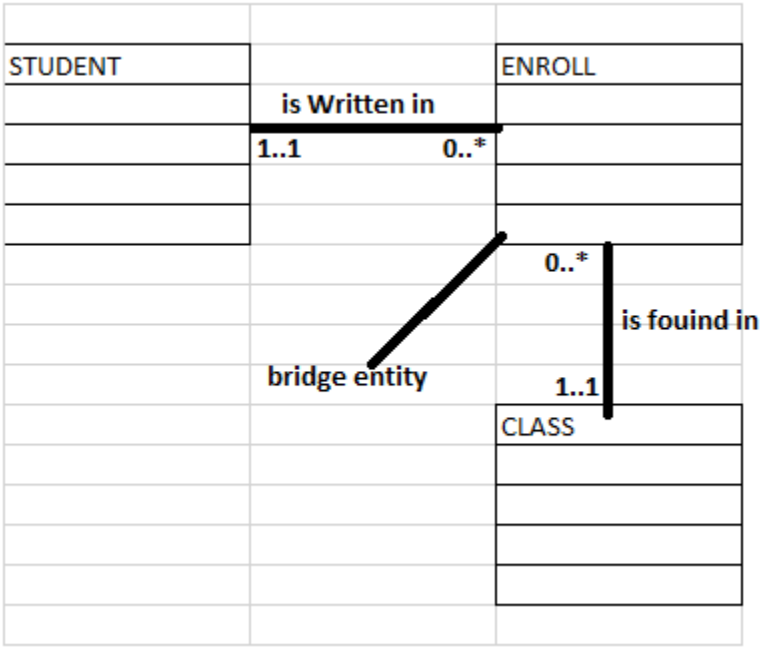
Student N:M enrolls Class

ERD:



Then if we introduce enroll entity which will use composite key combination of student primary and class primary and break up M:M relationship into 1:M and M:1 then Enroll entity is a bridge or composite entity.

ERD

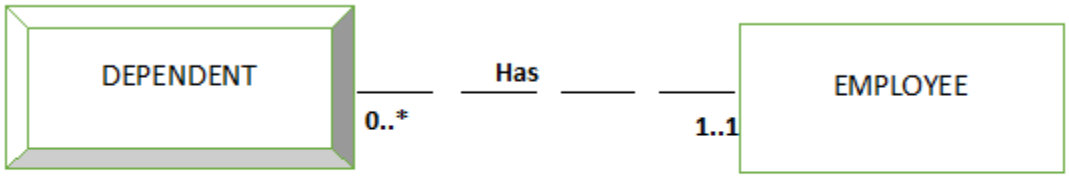


Relationship strength

- ❖ Relationship strength is based on the primary key of a related entity defined.
- ❖ Relationship strength define the strength of relationship between two entities.
- ❖ Relationship strength consist of two types which are:
    - Weak(non-identifying) relationship

- ❖ A weak relationship also known as non-identify
- ❖ Weak relationship exists if the primary key component of the child (related) entity is not generated from the primary key of parent entity.
- ❖ Weak relationship is represented by dashed relationship line.

Example of ERD



WEAK RELATIONSHIP

Example: give the following relational schema to determine relationship strength between employee and dependent entity schemas.

EMPLOYEE (EMP_**ID**, EMP_NAME, EMP_LNAME, EMP_DOB) AND DEPENDENT (EMP_ID**, DEP_ID**, DEP_NAME, DEP_DOB)

We can conclude that the relationship strength between two entities is weak relationship why, because the primary key of the child entity which is dependent table does not contain primary key of the parent entity. Since dependent table uses **DEP_ID** as a primary key not composite key.

- Strong (identifying) relationship

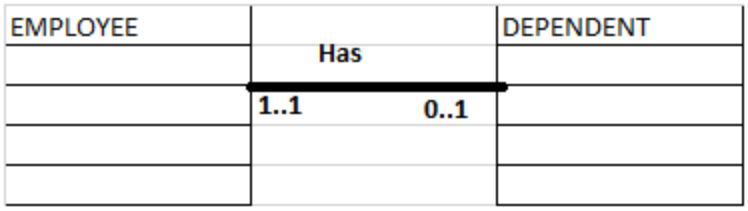
  - ❖ Strong relationship is also known as identify relationship
  - ❖ Strong relationship exists if the primary key of the related(child) table contains a primary component of the parent entity.
  - ❖ Strong relationship is represented by sold relationship line.

Example: considering the following relational schema to determine relationship strength between employee and dependent entity schemas.

EMPLOYEE (EMP_**ID**, EMP_NAME, EMP_LNAME, EMP_DOB) AND DEPENDENT (**EMP_ID, DEP_ID**, DEP_NAME, DEP_DOB)

We can conclude that the relationship strength between two entities is a strong relationship because the composite primary key of the child entity which is Dependent table contains the primary key component of the parent entity.

Example of ERD

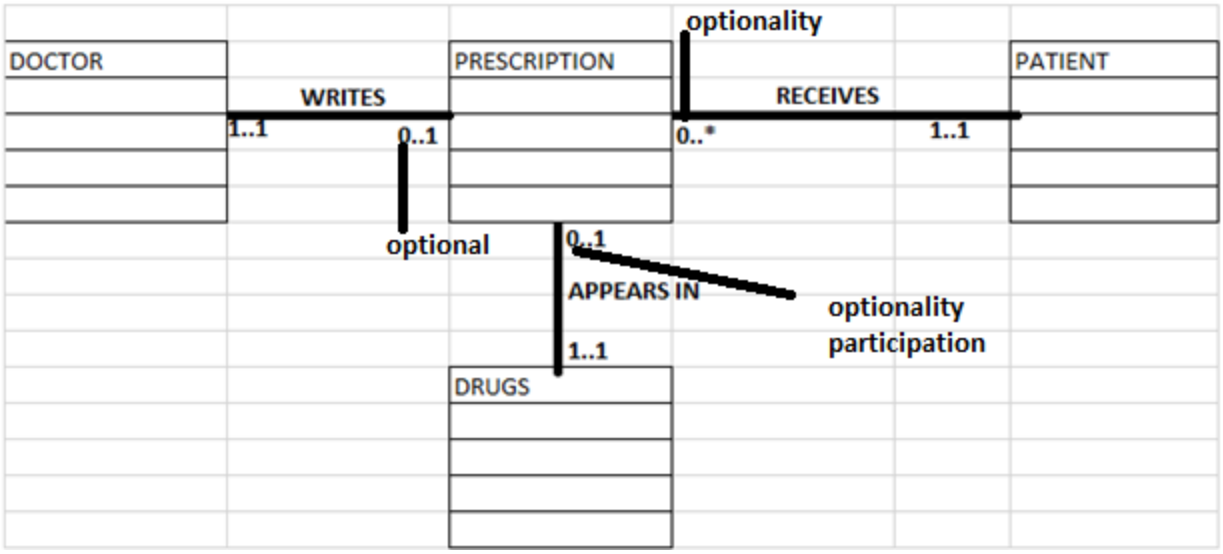


Example of strong relationship

ATTRIBUTE

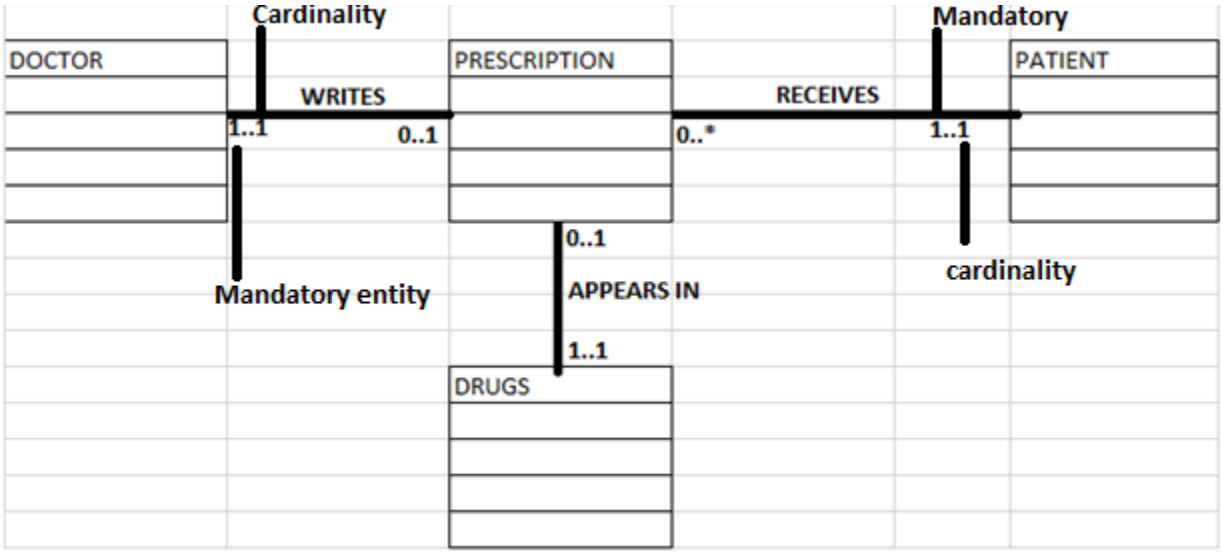
- Attribute Is the characteristic of an object or entity.
- Types of attributes:
  - Required and Optional attribute.
    - Required attribute is an attribute that must have the value, in short cannot be a null value.
    - Optional attribute is an attribute that does not require a value when a row generated.
  - Simple and composite attribute
    - Composite attribute is an attribute that can be further subdivided to yield additional attributes such as address can subdivide in to street, city, state and code.
    - Simple attribute is an attribute that cannot further subdivide such as age, sex, and marital status.
  - Single-valued attribute and multivalued attribute.
    - Single-valued attribute is an attribute that can only take a single value such as ID number always a person can be assigned to one ID, you cannot change ID number.
    - Multivalued attribute is an attribute that can take multiple values such as college degree, one person can have multiple qualification or telephone one person can be assigned to multiple phone number etc.
  - Derived attribute is an attribute that is generated from other attributes such as average, sum etc.
- Relationship participation
  - Participation in the relationship is either optional or mandatory
    - Optionality relationship participation.
      - Optionality participation means that one entity occurrence does not require a corresponding entity occurrence participation in the relationship.
      - Optional entity ranges from zero.
      - Optional entity represented by zero on the ERD.

Example:



- Mandatory relationship participation
  - Mandatory relationship participation means that one entity occurrence does require the corresponded entity occurrence participation in the relationship.
  - Mandatory entity the cardinality must ranges from one.

Example:



Relationship degrees

- Relationship degree indicates the number of entities involved in the relationship.
- Type of relationship degrees:

- Unary relationship
  - Unary relationship is also known as recursive relationship.
  - Unary relationship exists when one entity has a relationship with itself

Example:

Consider the following business rules and draw entity relationship diagram using CHEN model.

Each employee manages many employees.

Diagram

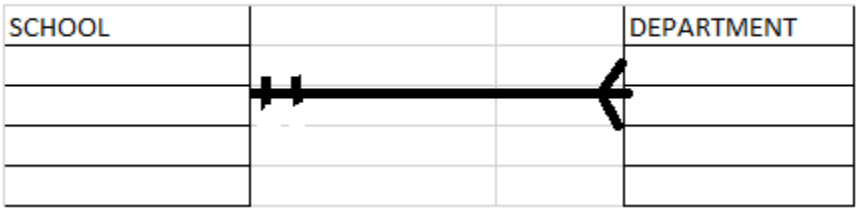
- Binary relationship
  - Binary relationship exists when two entities involved/ associated in the relationship.
  - Binary relationship is almost used in data modeling.

Example:

Consider the following business rules and draw entity relationship diagram using CROW's foot notation

Each school composed of several departments.

Diagram

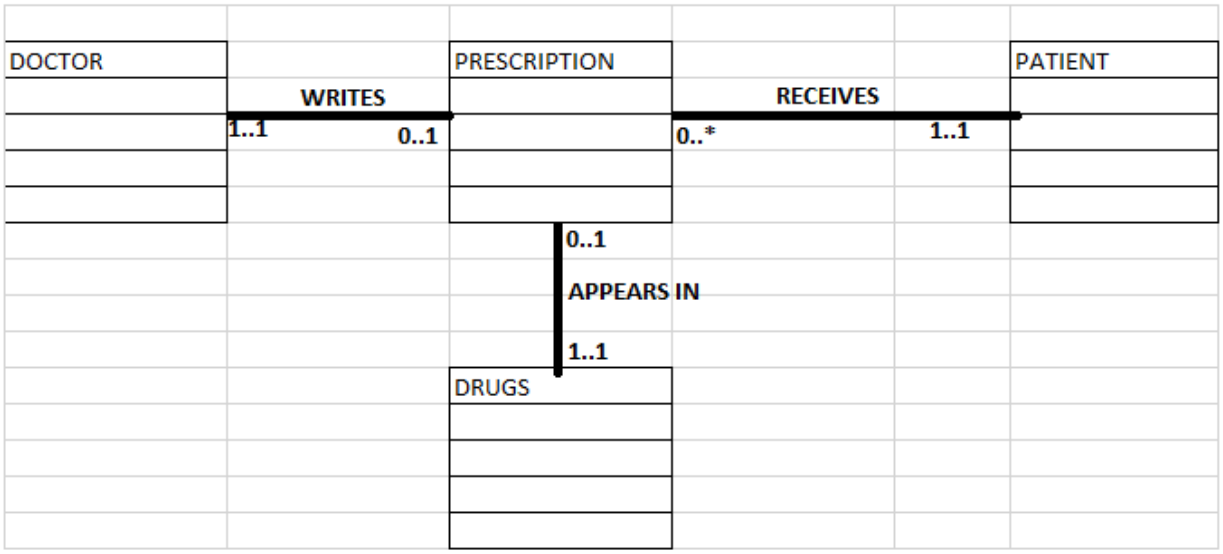


- Ternary relationship
  - Ternary relationship exists when more than two entities associated in the relationship.

Example:

Consider the following business rules and draw entity relationship diagram using UML notation.

A doctor writes one or more prescription's and a patient may receive one or more prescription, each prescription may include one drugs and a drug may appear in one or more description.



Advance data modelling

The extended entity relationship model

- The extended relationship model is referred as the enhanced entity relationship model is result of adding some sematic constructs to the original entity relationship model.
- The diagram used in this model is called enhanced entity relationship diagram.
- Modification or addition sematic is to add the subtype and supertype entities.

Entity supertype

- Entity supertype is an entity that minimize the difference between entities.

Example: Employee is a supertype entity because we can subdivide employee into subtype entities such as teacher, principal, director and cleaner etc.

Entity subtype

- Subtype entity is an entity that maximize difference between entity.
- Example: teacher and principal is example or type of employee but they differ with job descriptions then are called subtype.

Specialization hierarchy

- Specialization is the top-down process of identifying lower-level, more specific entity subtypes from higher level entity supertype.
- Specialization is based on grouping unique characteristics and relationship of the subtype entity

Generalization hierarchy

- Generalization is the bottom -up process of identifying higher level, more generic entity supertype from lower – level entity subtype.
- Generalization is base on grouping common characteristics and relationship of the entity subtype.

Inheritance

- The property of the inheritance enables an entity subtype to inherit the attributes and relationships of the supertype entity.

Completeness constraint

- The completeness constraints specify whether each entity supertype occurrence must also be a member of at least one subtype.
- The completeness constrains is divided into partial completeness constraint and total completeness constraint.

Partial completeness

- Partial completeness means that not every supertype occurrence is a member of a subtype.
- Partial completeness denoted by a circle over a single line.

Total completeness

- Total completeness means that every supertype occurrence is a member of a subtype.
- Total completeness is donated by a circle over a double line.

Disjoint constraints

- Disjoint subtypes are subtype that contain unique subset of the supertype entity set.
- Disjoint constraint also known as no overlapping subtype.

Overlapping constraint

- Overlapping subtypes are subtypes that contains nonunique subset of the supertype entity set.

Example done in class

The flyRight aircraft maintenance (FRAM) division of the flyRight company (FRC) Performs all maintenance for FRC aircraft. Produce a data model segment that reflects the following business rules:

- All mechanics are FRC employee, not all employees are mechanics
- Some mechanics are specialized in engine (EN) maintenance. Some mechanics are specialized in airframe (AF) maintenance. Some mechanics are specialized in avionic (AV) maintenance. (AV are the electronic components of an aircraft that are used in communication and navigation.) all mechanics take periodic refresher course to stay current in their areas of expertise. FRC track all course taken by each mechanics- date, course type, certification(Y/N), and performance.
- FRC keeps a history of the employment of all mechanics. History includes the date hired, date promoted, date terminated, and so on. (note: the “and so on” components is, of course, not

areal -world requirement. Stead, it been used here to limit the number of attributes you will show in your design.)

Normalization

- Normalization is the process of evaluating and correcting table structure to minimize data redundancy, whereby reducing the likelihood of data anomaly
- Normalization process works under five stages normal forms namely:
    - First normal form
    - Second normal form.
    - Three normal form
    - Boyce-codd normal form

Dependency diagram

Dependency diagram consist of three types of dependencies namely:

- **Full functional dependency**
    - Functional dependency exists when the primary or composites primary key able to define all non-primary key attributes in the table.

Example: consider following rational schema for employee table:

Employee (employee_code, employee_name, employee_address, employee_salary), If all attributes are full depending on employee_code then they create a dependency called functional dependency. Employee_code→ employee_name, employee_address, employee_salary

- **Partial dependency**
    - Partial dependency exists if non-primary key attributes are fully depending on the portion of the composite primary key.

Example consider the following rational schema: dependent (**employee_code**, **dependent_num**, dependent_name, dependent_age, employee_comapany_name). The table use a composite primary key then if the dependent number can able to define the values of dependent_name, dependent_age then we say dependent name and dependent age attributes are fully depending on de pendent number which is the portion of the composite key then they create a partial dependency.

- **Transitive dependency**
    - Transitive dependency exists when a non-primary key attribute defines another non-primary key attribute.

Example: consider the following rational schema: Employee (employee_code, employee_name, employee_address, employee_salary, job_class), if job_class attributes can able to define employee salary which is also non-primary key attribute then they create a relationship called a transitive dependency.

Job_class→ employee_salary both are non-primary key attributes.

First normal form

- Table is in the first normal form if there is no repeating groups

Converting the table in first normal form and drawing dependency diagram

STEP 1: Eliminate the repeating group

- First step is to check whether the table contains repeating groups then if does contain group, to eliminate groups, eliminate null values by make sure that each cell contains appropriate data value.

Example consider the table below:

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_LNAME	JOB_CLASS	CHG_HOUR	HOUR
15	EVERGREEN	103	JUNE E ARBOUGH	ELECT ENGINEER	R67.55	23.8
		101	JOHN G NEWS	DATABASE DESIGNER	R82.95	19.4
		105	ALICE K JOHNSON	DATABASE DESIGNER	R82.96	35.7
		106	WILLIAM SMITHFIE	PROGRAMMER	R26.66	12.6
		102	DAVID H SENIOR	SYSTEM ANALYST	R76.43	23.8
18	AMBER WAVE	114	ANNELISE JONES	APPLICATION DESIGNER	R38.00	24.6
		118	JAMES J FROMMER	GENERAL SUPPORT	R14.50	45.3
		104	ANNE K RAMORAS	SYSTEM ANALYST	R76.43	32.4
		112	DARLENE M SMITH	DSS ANALYST	R36.30	44
22	ROLLING TIDE	105	ALICE K JOHNSON	DATABASE DESIGNER	R82.96	35.7
		104	ANNE K RAMORAS	SYSTEM ANALYST	R76.43	48.4
		113	DELBERT JOENBRO	APPLICATION DESIGNER	R38.00	23.6
		106	WILLIAM SMITHFIE	PROGRAMMER	R26.66	12.6
25	STARFLIGHT	107	MARIA D	PROGRAMMER	R26.66	12.8
		115	TRAVIS B. BAWAN	SYSTEM ANALYST	R76.43	45.8
		101	JOHN G NEWS	DATABASE DESIGNER	R82.95	56.3
		114	ANNELISE JONES	APPLICATION DESIGNER	R38.00	33.1
		118	JAMES J FROMMER	GENERAL SUPPORT	R14.50	30.5
		112	DARLENE M SMITH	DSS ANALYST	R36.30	44

If can read the content of the table you will see that data is grouped into a proj number and proj name so ungroup the data, eliminate null values by repeating the value of the project number and project name as show below:

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_LNAME	JOB_CLASS	CHG_HOUR	HOUR
15	EVERGREEN	103	JUNE E ARBOUGH	ELECT ENGINEER	R67.55	23.8
15	EVERGREEN	101	JOHN G NEWS	DATABASE DESIGNER	R82.95	19.4
15	EVERGREEN	105	ALICE K JOHNSON	DATABASE DESIGNER	R82.96	35.7
15	EVERGREEN	106	WILLIAM SMITHFIE	PROGRAMMER	R26.66	12.6
15	EVERGREEN	102	DAVID H SENIOR	SYSTEM ANALYST	R76.43	23.8
18	AMBER WAVE	114	ANNELISE JONES	APPLICATION DESIGNER	R38.00	24.6
18	AMBER WAVE	118	JAMES J FROMMER	GENERAL SUPPORT	R14.50	45.3
18	AMBER WAVE	104	ANNE K RAMORAS	SYSTEM ANALYST	R76.43	32.4
18	AMBER WAVE	112	DARLENE M SMITH	DSS ANALYST	R36.30	44
22	ROLLING TIDE	105	ALICE K JOHNSON	DATABASE DESIGNER	R82.96	35.7
22	ROLLING TIDE	104	ANNE K RAMORAS	SYSTEM ANALYST	R76.43	48.4
22	ROLLING TIDE	113	DELBERT JOENBRO	APPLICATION DESIGNER	R38.00	23.6
22	ROLLING TIDE	106	WILLIAM SMITHFIE	PROGRAMMER	R26.66	12.6
25	STARFLIGHT	107	MARIA D	PROGRAMMER	R26.66	12.8
25	STARFLIGHT	115	TRAVIS B. BAWAN	SYSTEM ANALYST	R76.43	45.8
25	STARFLIGHT	101	JOHN G NEWS	DATABASE DESIGNER	R82.95	56.3
25	STARFLIGHT	114	ANNELISE JONES	APPLICATION DESIGNER	R38.00	33.1
25	STARFLIGHT	118	JAMES J FROMMER	GENERAL SUPPORT	R14.50	30.5
25	STARFLIGHT	112	DARLENE M SMITH	DSS ANALYST	R36.30	44

So now the table is in the first normal form because there are no repeating groups.

Step 2 identify the primary key attributes

The first step to identify key check all attributes with the characteristics of the primary key then after check whether the values repeating or not, if the values are repeating compare with other columns whether the values are same or not. If are not means the table uses a composite primary key not a primary key.

Example let consider the table above.

The column which qualifies to be a primary key is a proj_num from the table but alone cannot return a unique row, for example if we take 15 will return first five rows and they are not the same then is clearly that we cannot use proj_num be to a primary key, then we check next attribute which is emp_num again employee number is returning multiple rows so now we are forest to combine proj_num and emp_num to key of the table. now example if we take combination between proj_num 15 and emp_num 101 will return only records which is

15	EVERGREEN	101	JOHN G NEWS	DATABASE DESIGNER	R82.95	19.4

Then we can use the combination of PROJ_NUM and EMP_NUM to be our key.

Identify the dependent attributes

From the example above

- PROJ_NUM and EMP_NUM define the remaining attribute the is a full functional dependency
- Then we look attribute which are only depending on PROJ_NUM only, then we choose two rows where the value of the PROJ_NUM is the same and compare other values, the values are the same that define that attribute is full depending on PROJ_NUM

EXAMPLE: Consider two rows in the table where the value of PROJ_NUM is the same then compare values of other attributes

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_LNAME	JOB_CLASS	CHG_HOUR	HOUR
15	EVERGREEN	103	JUNE E ARBOUGH	ELECT ENGINEER	R67.55	23.8
15	EVERGREEN	101	JOHN G NEWS	DATABASE DESIGNER	R82.95	19.4

If can check from table above emp_num, emp_lname, job_class chg_hour and hour the values are different, then that tells us that those attributes are not fully depending on PROJ NUM, only PROJ_NAME is fully depending on PROJ_NUM because the values are the same. Then we have PROJ_NUM→ PROJ_NAME which is a partial dependency. Same procedure for EMP_NUM firstly we check value of employee number which is repeating and compare other values

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_LNAME	JOB_CLASS	CHG_HOUR	HOUR
15	EVERGREEN	101	JOHN G NEWS	DATABASE DESIGNER	R82.95	19.4
25	STARFLIGHT	101	JOHN G NEWS	DATABASE DESIGNER	R82.95	56.3

If you can check the value of proj_nu, proj_name and hour are totally different means that those attributes are not fully depending on EMP_NUM but remaining attribute are fully depending on EMP_NUM. Rational schema: EMP_NUM → EMP_LNAME, JOB_CLASS, CHG_HOUR which is a partial dependency.

Consider non-primary key attribute to define transitive dependency lets consider the table:

PROJ_NAME	EMP_LNAME	JOB_CLASS	CHG_HOUR	HOUR
EVERGREEN	JUNE E ARBOUGH	ELECT ENGINEER	R67.55	23.8
EVERGREEN	JOHN G NEWS	DATABASE DESIGNER	R82.95	19.4
EVERGREEN	ALICE K JOHNSON	DATABASE DESIGNER	R82.96	35.7
EVERGREEN	WILLIAM SMITHFIE	PROGRAMMER	R26.66	12.6
EVERGREEN	DAVID H SENIOR	SYSTEM ANALYST	R76.43	23.8
AMBER WAVE	ANNELISE JONES	APPLICATION DESIGNER	R38.00	24.6
AMBER WAVE	JAMES J FROMMER	GENERAL SUPPORT	R14.50	45.3
AMBER WAVE	ANNE K RAMORAS	SYSTEM ANALYST	R76.43	32.4
AMBER WAVE	DARLENE M SMITH	DSS ANALYST	R36.30	44
ROLLING TIDE	ALICE K JOHNSON	DATABASE DESIGNER	R82.96	35.7
ROLLING TIDE	ANNE K RAMORAS	SYSTEM ANALYST	R76.43	48.4
ROLLING TIDE	DELBERT JOENBRO	APPLICATION DESIGNER	R38.00	23.6
ROLLING TIDE	WILLIAM SMITHFIE	PROGRAMMER	R26.66	12.6
STARFLIGHT	MARIA D	PROGRAMMER	R26.66	12.8
STARFLIGHT	TRAVIS B. BAWANG	SYSTEM ANALYST	R76.43	45.8
PROJ_NAME	EMP_LNAME	JOB_CLASS	CHG_HOUR	HOUR
EVERGREEN	JOHN G NEWS	DATABASE DESIGNER	R82.95	19.4
STARFLIGHT	JOHN G NEWS	DATABASE DESIGNER	R82.95	56.3
STARFLIGHT	ANNELISE JONES	APPLICATION DESIGNER	R38.00	33.1
STARFLIGHT	JAMES J FROMMER	GENERAL SUPPORT	R14.50	30.5
STARFLIGHT	DARLENE M SMITH	DSS ANALYST	R36.30	44

The table does not include composite primary key attribute, the key is to identify the relationship that exist between non-primary key attribute. We apply the same procedure of choose one value which is repeating and compare values of other attributes, firstly we will choose evergreen, when we compare we find that all values of other attribute are different which means that there is no attribute which is fully depending on PROJ_NAME, but if we check for JOB CLASS we will notice that all employees employed as database design the earn same salary also programmer they earn the same salary that tells us that JOB_CLASS define CHG_HOUR. Then create the relationship called transitive dependency.

Dependency diagram

Second normal form

- The table said to be in the 2nd normal form:
    a. The table is in first normal form.
    b. The table must not contain partial dependency.

Conversion to second normal form

- Converting to 2an normal form is done only when the first normal form has a composite primary key.

- If the table is in first normal form and has a single attribute primary key, then the table is in second normal.

Step 1: write each key component on a separate line

Example: considering the example above we will separate proj_num and emp_num and we will have:

PRO_NUM

EMP_NUM

STEP 2: Identify the dependent attribute

To identify the dependent attribute, we follow the partial link from dependency diagram in the first normal form, then we will have the following schema:

PROJ_NUM→PROJ_NAME……1

EMP_NUM → EMP_NAME, JOB_CLASS, CHG_HOUR……2

But from the original table we are left with hours that means if we need to know the values of attribute hour we need the combination of the PROJ_NUM and EMP_NUM. so now we are forest to combine PROJ_NUM and EMP_NUM back so that we cannot loses hour attribute, then we will have:

PROJ_NUM, EMP_NUM→ HOURS……3 from one table now we generated three table which are in second normal form.

Rational schemas of second normal form tables:

1. PROJ_NUM→PROJ_NAME
2. EMP_NUM→ EMP_NAME, JOB_CLASS, CHG_HOUR
3. EMP_NUM, PROJ_NUM → HOURS

DEPENDENCY DIAGRAM IN 2ND NF

Third normal form

- The table is said to be in third normal form if and only if:
    a. The table is in second normal form.
    b. The table does not contain transitive dependency

Conversion to third normal form

Step 1: identify non-primary attribute determinate

To identify new determinate, you follow the link for transitive dependency from the tables in second normal form. From the tables above, we will identify the following determinate which exist between job class and chg hour, the relational schema:

JOB_CLASS → CHG_HOUR.

Then from employee table we are left with Emp_Name but we must include JOB_CLASS attribute to create a foreign key to show that the tables are related. Then we will have the following tables:

JOB_CLASS→ CHG_HOUR

EMP_NUM→EMP_NAME, JOB_CLASS

EMP_NUM, PROJ_NUM→ HOURS

PROJ_PROJ_NAME

All tables are in the third normal form because the tables are in the second normal form and there is no transitive dependency

Dependency diagram in third normal form.

Example:

DONE IN CLASS