JAVA FULL STACK PROGRAM

Session: SQL Day 1

OUTLINE

* Data Modeling

* ER Diagram

* Normalization

* Database

* SQL

* Constraints

DATA MODELING

- Data Model is the process of analyzing business requirement, designing and creating physical instance (a plan or blueprint) for the DB or DW

- It is a stage in SDLC (Software Development Life Cycle)

- It also impacts the user interface

Relational database Terminology

SQL term	Relational database term	Description
Row	Tuple or record	A data set representing a single item
Column	Attribute or Field	A labeled element of a tuple, e.g. "Address" or "Date of birth"
Table	Relation or Base relvar	A set of tuples sharing the same attributes; a set of columns and rows
View or Result set	Derived relvar	Any set of tuples; a data report from the RDBMS in response to a query

Relational database Terminology

DATA MODELING TERMINOLOGY

- Entity

  - An item that can exist independently or uniquely identified

- Attribute

  - Column label, name

- Domain

  - Set of valid values for an attribute

- Relationship

  - How entities relate

- Degree

  - How many entities in a relationship

- Cardinality

  - Measure of participation

KEYS USED IN DATA MODELING

- Superkey

  - One or multiple attributes that can uniquely identify a tuple

- Candidate Key / Unique Key

  - Minimal superkey

  - No proper subset of candidate key can uniquely identify a tuple

- Primary Key

  - Selected candidate key

  - One per table

  - Usually ID column (auto increment integer)

- Alternate key

  - All candidate keys that are not primary key

- Surrogate Key

  - Unique identifier internally generated by the system for an entity in the modeled world or an object in the database

- Natural Key

  - Unique key that is formed of attributes that already exist in the real world

  - A natural key is a candidate key and is therefore a functional determinant for all attributes in a relation

    - eg. Purchase Order number

- Composite Key

  - Superkey with two or more attributes

KEYS USED IN DATA MODELING

Candidate Key

Primary Key

Alternate Key

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

Address

🔑	AddressID
	StreetNumber
	StreetName
	City
	State
	ZipCode
	ModifiedDate

	AddressID	StreetNumber	StreetName	City	State	ZipCode	ModifiedDate
1	1	123	6th St.	Melbourne	FL	32904	2018-04-11 20:02:41.637
2	2	71	Pilgrim Avenue	Chevy Chase	MD	20815	2018-04-11 20:02:41.647
3	3	70	Bowman St.	South Windsor	CT	06074	2018-04-11 20:02:41.647
4	4	4	Goldfield Rd.	Honolulu	HI	96815	2018-04-11 20:02:41.647
5	5	44	Shirley Ave.	West Chicago	IL	60185	2018-04-11 20:02:41.650
6	6	514	S. Magnolia St.	Orlando	FL	32806	2018-04-11 20:02:41.650

Surrogate Key

Suit	Value	No. times played
Hearts	Ace	5
Diamonds	Three	2
Hearts	Jack	3
Clubs	Three	5
Spades	Five	1

One is not enough to identify, but both combined make a unique value

Employee

	FirstName
	LastName
	BirthDate
🔑	SSN
	AddressID
	HireDate
	ModifiedDate
	Notes

|	FirstName	LastName	BirthDate	SSN	AddressID	HireDate	ModifiedDate	Notes
1	Jim	Smith	1978-03-21	234-24-2343	1	2010-02-17	2010-12-31 00:00:00.000	
2	Tammy	Jones	1994-09-21	346-73-1944	2	2011-04-27	2015-12-31 00:00:00.000	
3	Sam	Donald	2001-08-17	487-14-3329	5	2014-12-07	2017-12-31 00:00:00.000	
4	Mike	MacDonald	1999-06-28	654-81-8273	3	2009-02-11	2010-12-31 00:00:00.000	
5	Stephanie	Shepherd	1998-06-02	834-94-9941	6	2013-01-01	2013-12-31 00:00:00.000	
6	Jennifer	Bowles	1994-12-01	943-91-4733	4	2015-12-22	2016-12-31 00:00:00.000	

Natural Key

CARDINALITY & DEGREE

- The number of times the entity participates in the relationship

  - One-to-One

    - One element in entity A may only link to one element in entity B and vice versa

  - One-to-Many

    - One element in entity A may link to many elements in entity B but one element in entity B may only link to one element in entity A

  - Many-to-One

    - Reverse A and B in one-to-many

  - Many to Many

    - One element in entity A may link to any number of elements in entity B and vice versa

- Cardinality != Degree

  - Degree -- This is the number of entities involved in the relationship and it is usually 2 (binary relationship) however Unary and higher degree relationships can be exists.
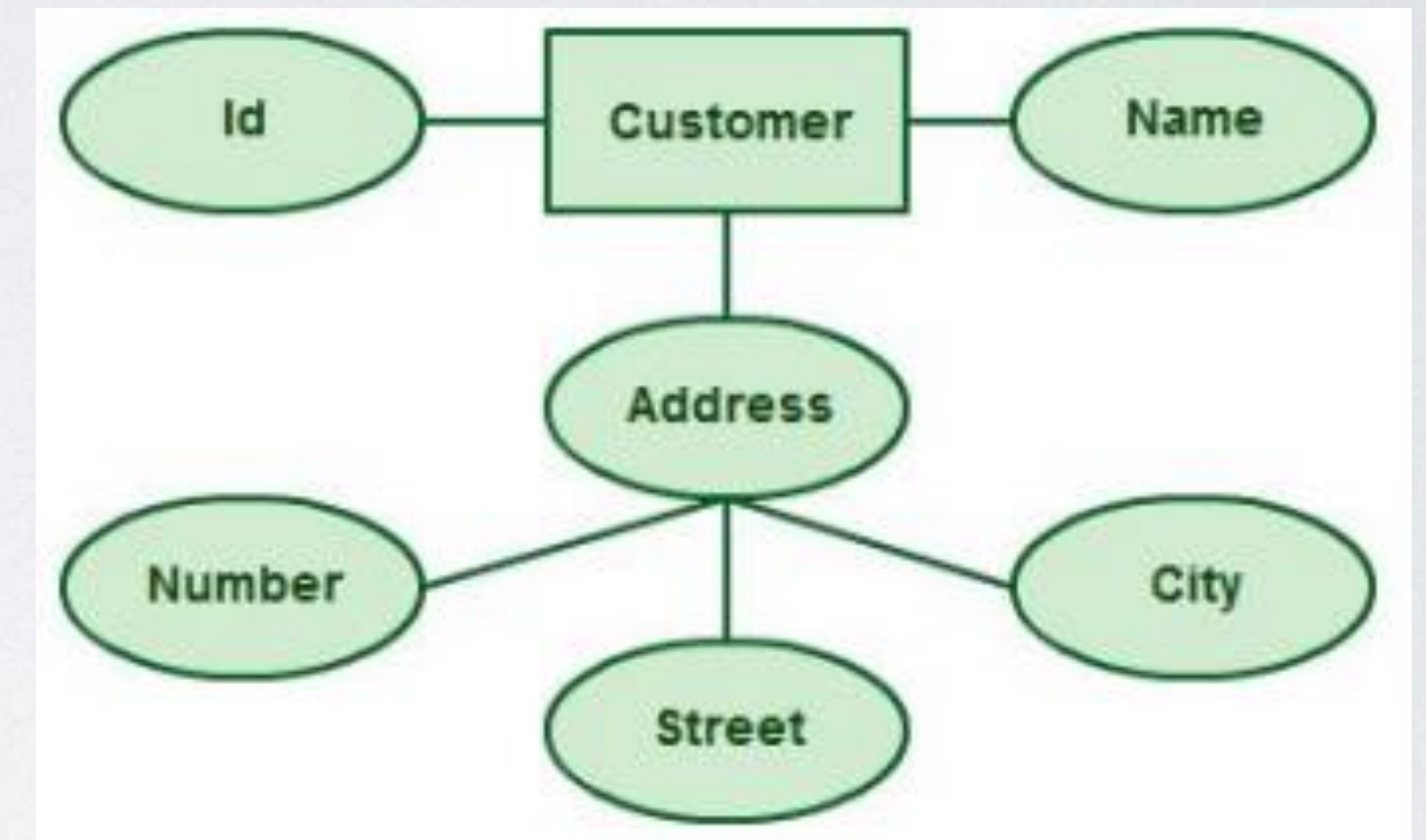
CARDINALITY & DEGREE

CARDINALITY & DEGREE

Unary Relationship
(degree 1)

Ternary Relationship
(degree 3)

Binary Relationship
(degree 2)
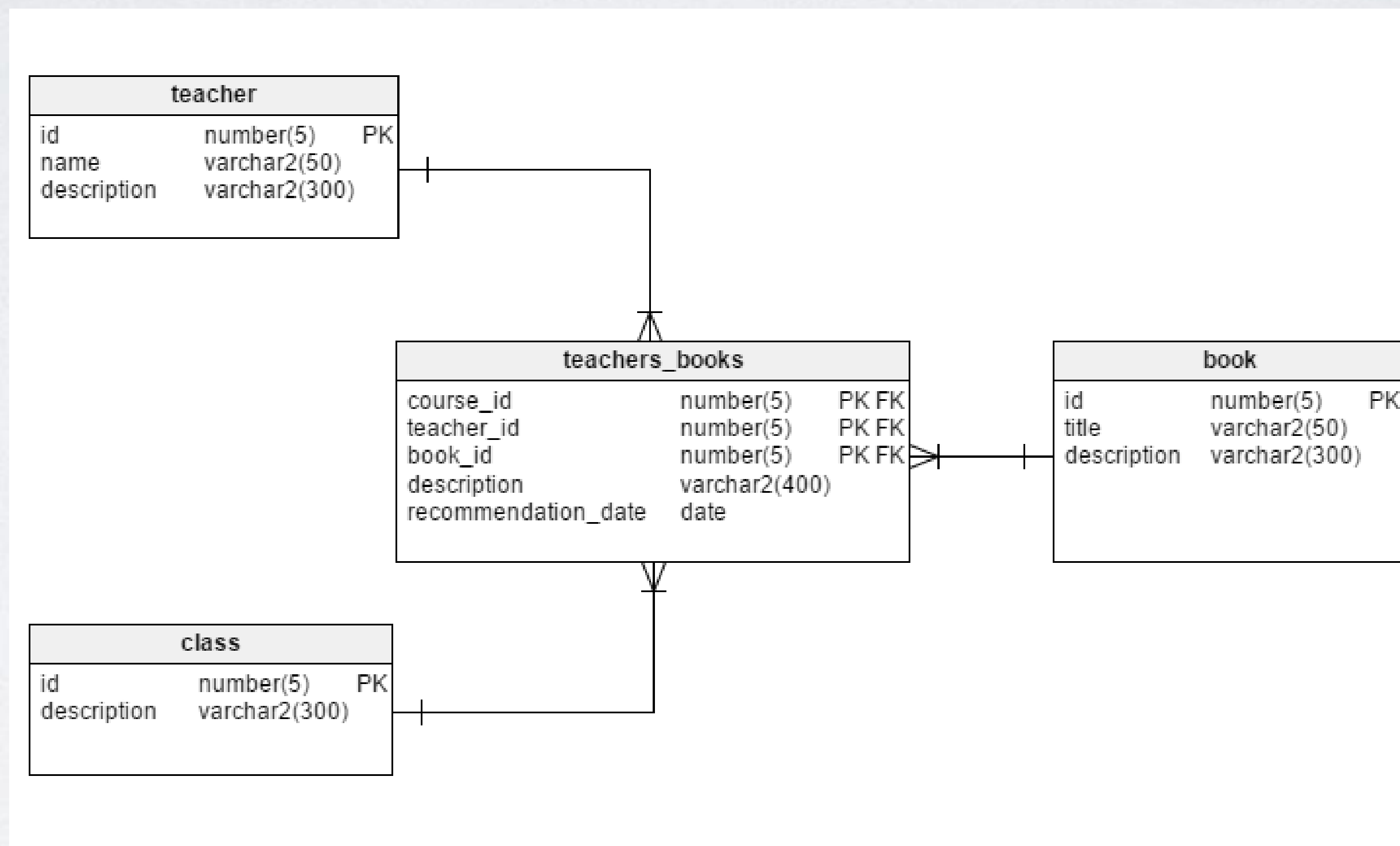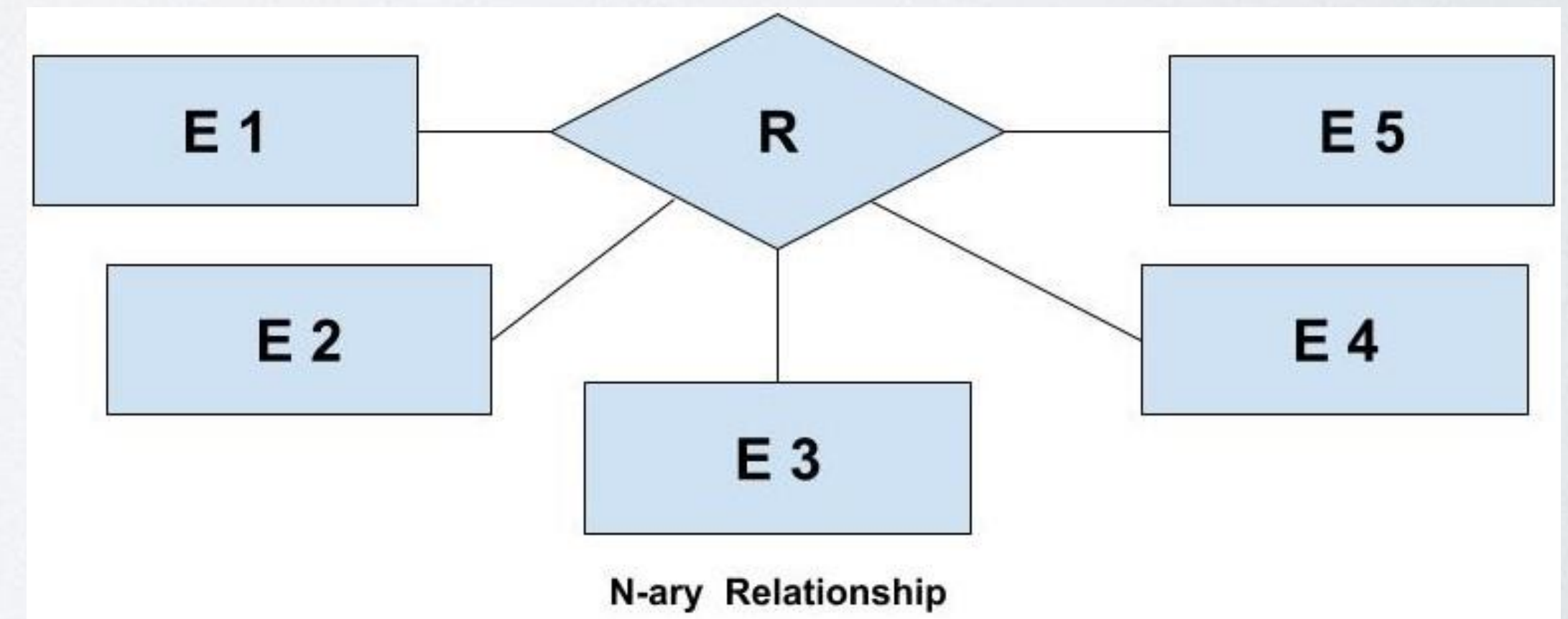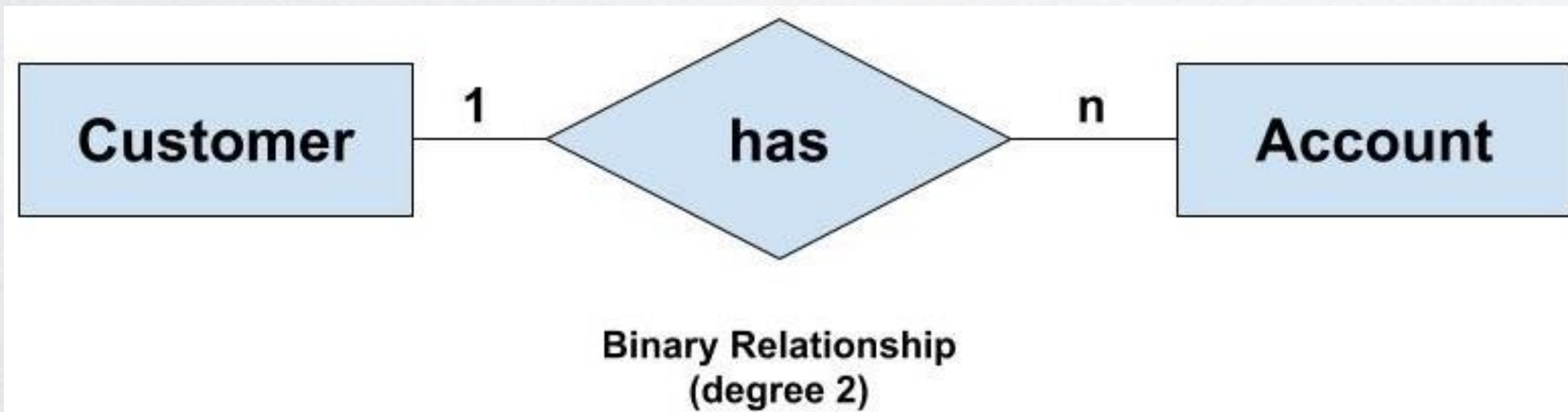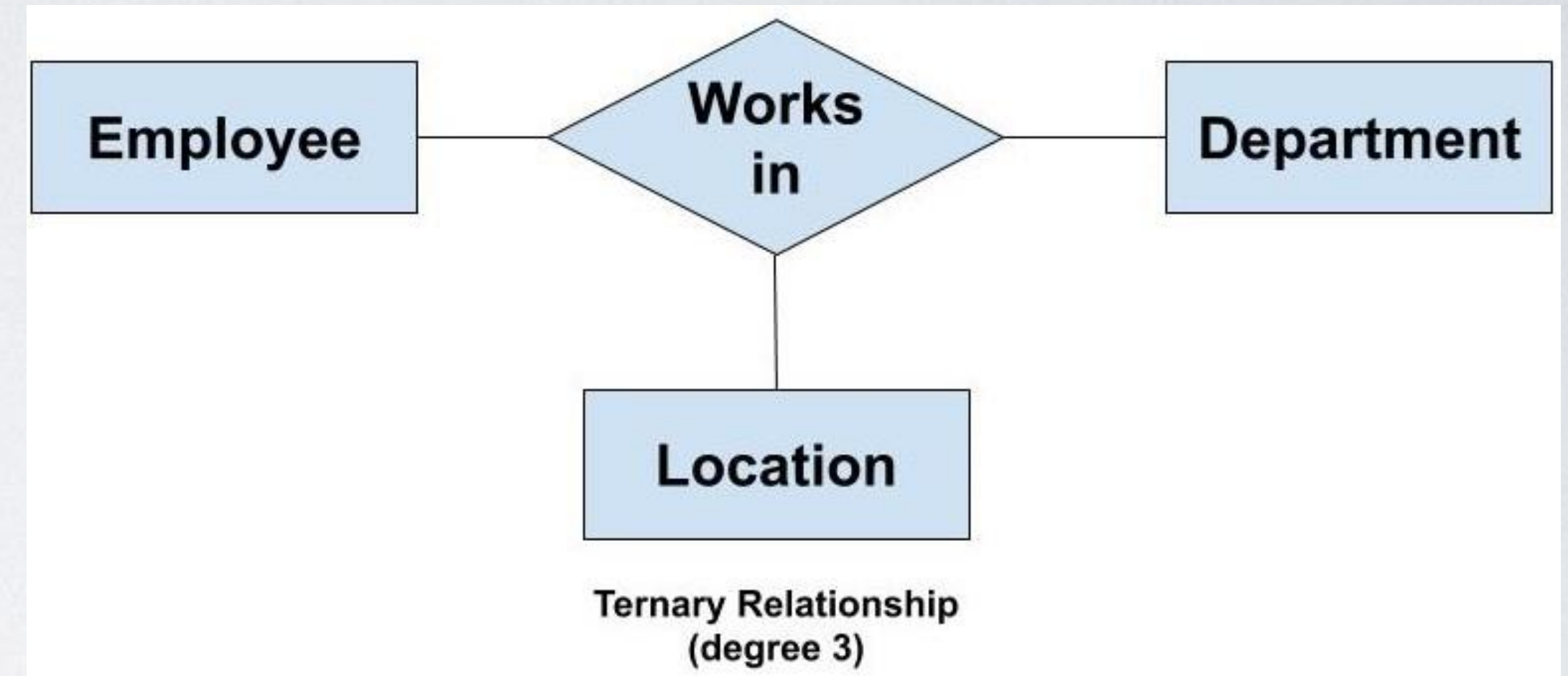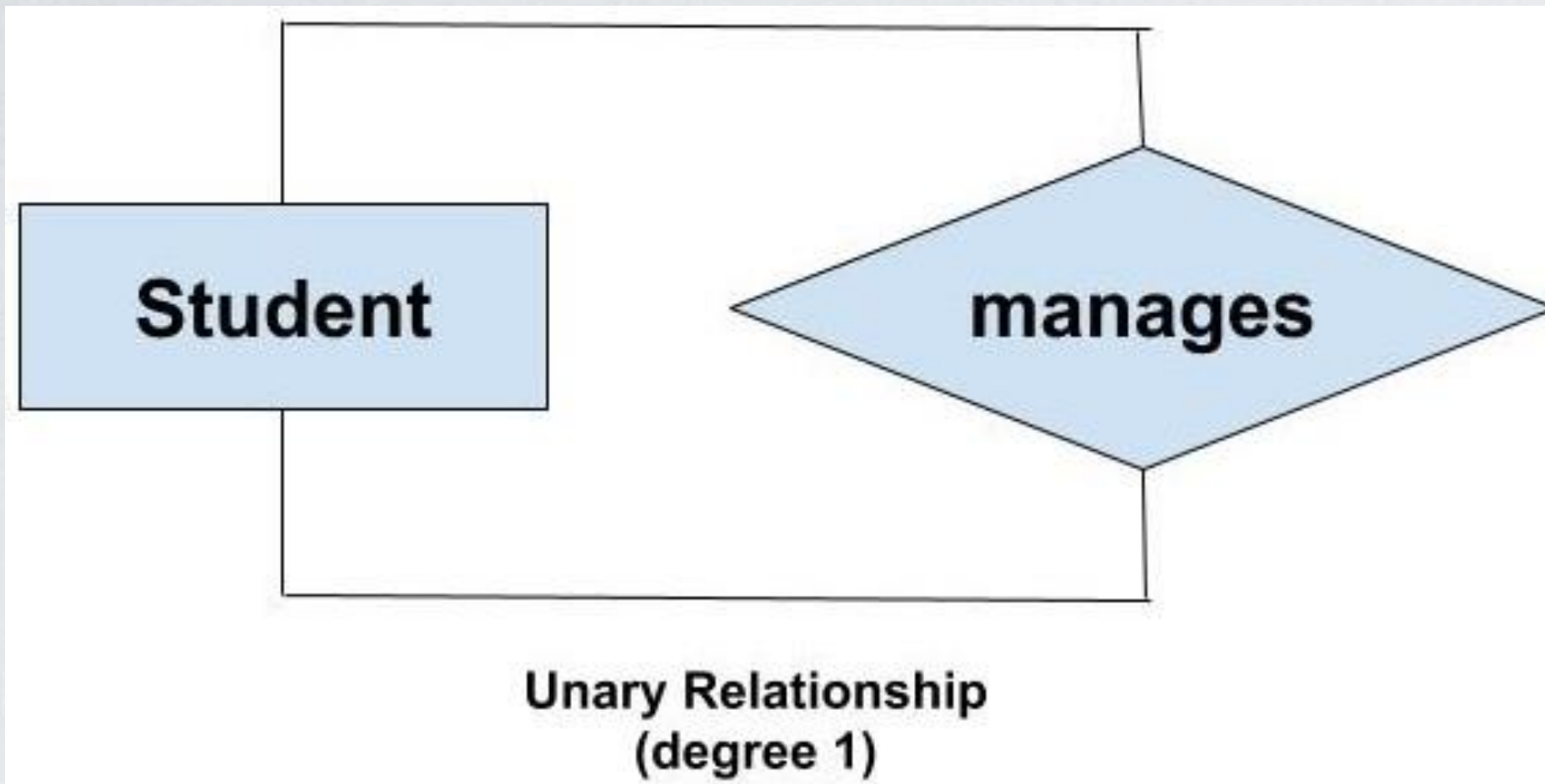
N-ary Relationship

ENTITY-RELATIONSHIP DIAGRAM
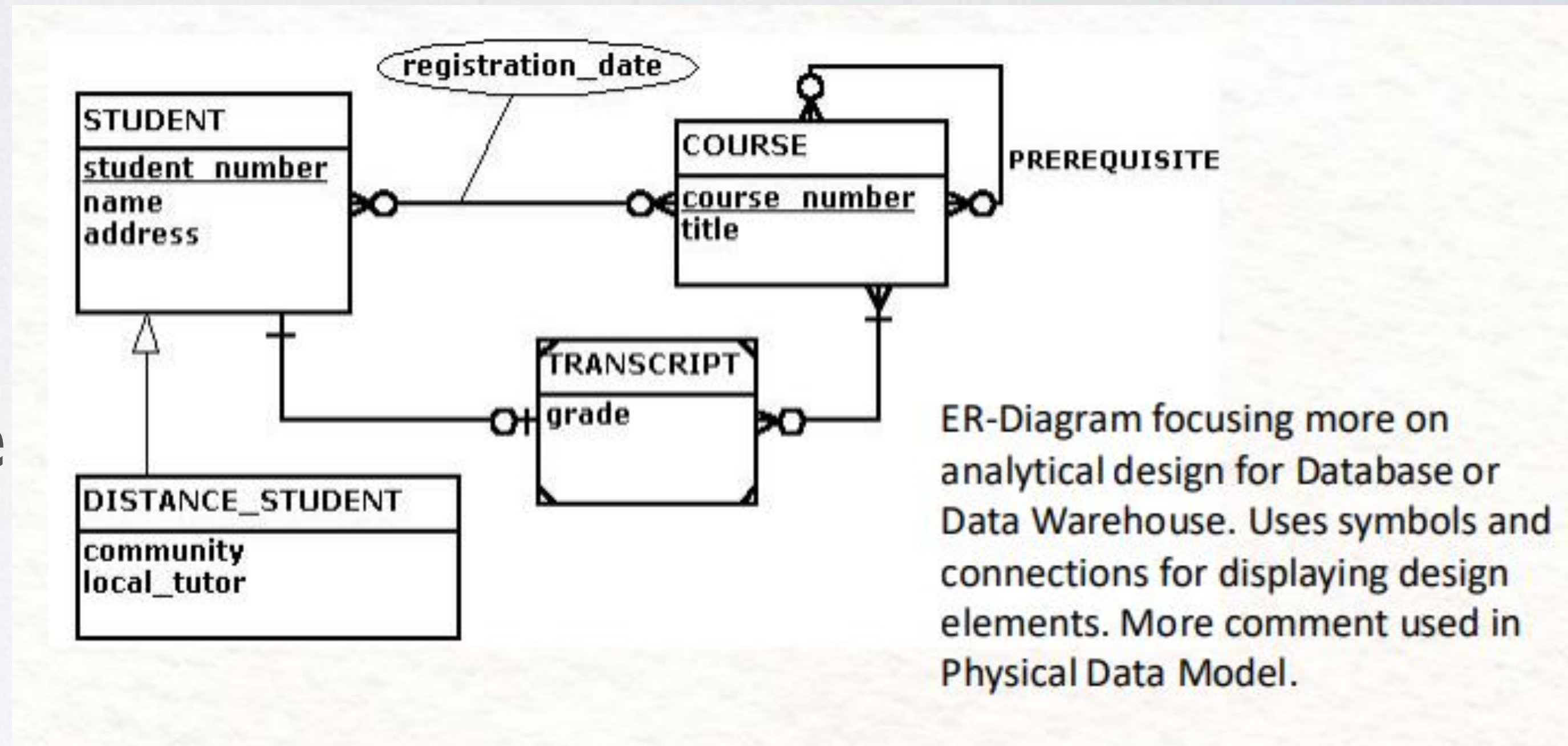
- ER-Diagram

  - Entity Relationship Diagram

  - Used to Create or Design a

  Blueprint of the Database or Data Warehouse
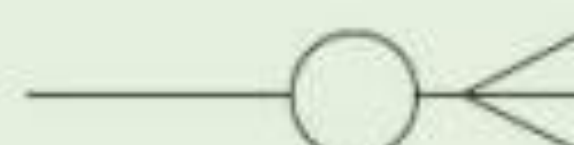
- Design Entities, Attributes and show Relationships

CROW'S FOOT NOTATION

- Connection Symbols display Relationships

- Entity and Attributes in Table like format

ER-Diagram focusing more on analytical design for Database or Data Warehouse. Uses symbols and connections for displaying design elements. More comment used in Physical Data Model.
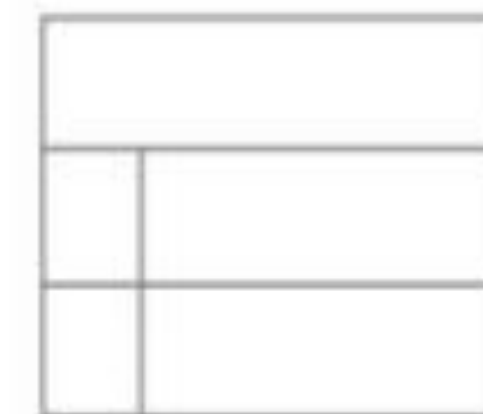
CROW'S FOOT NOTATION

CROW'S FOOT NOTATION
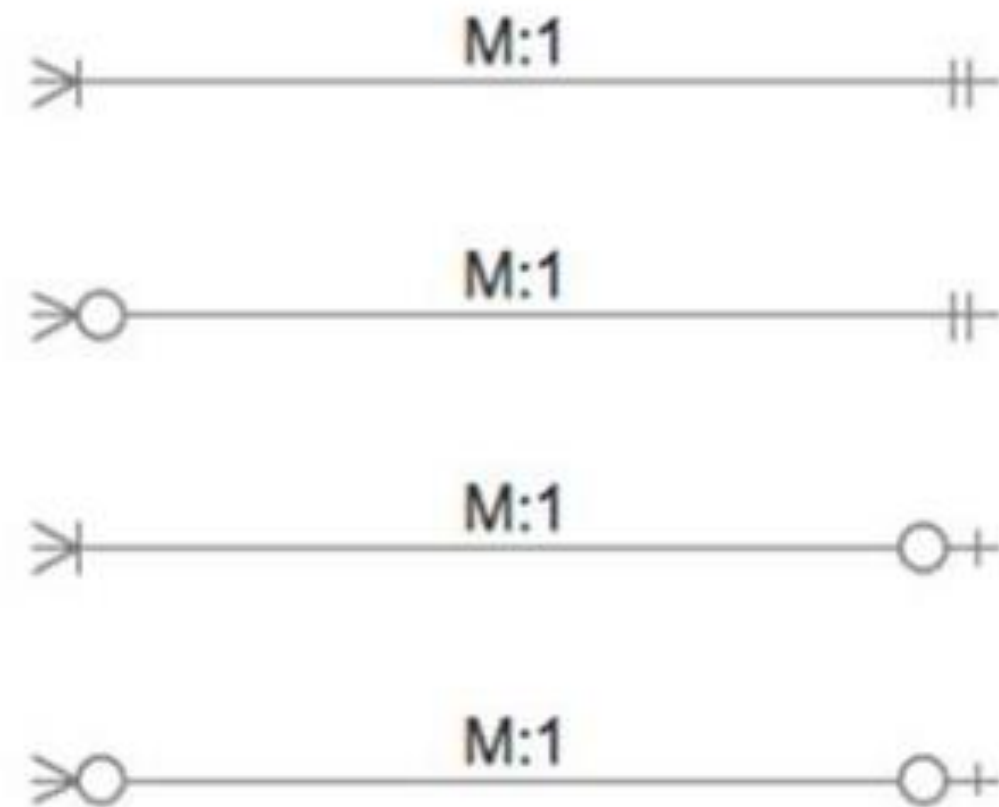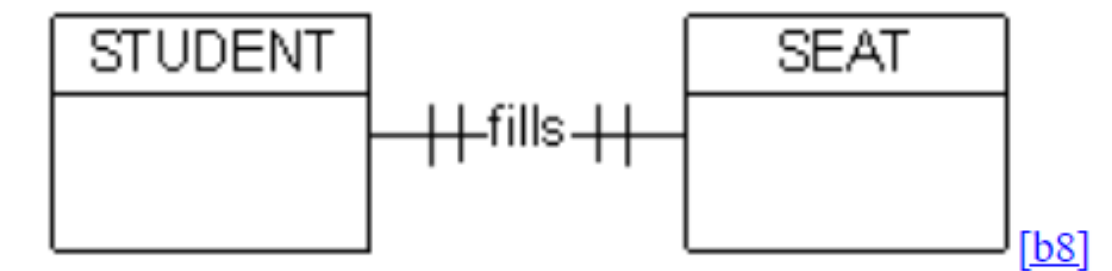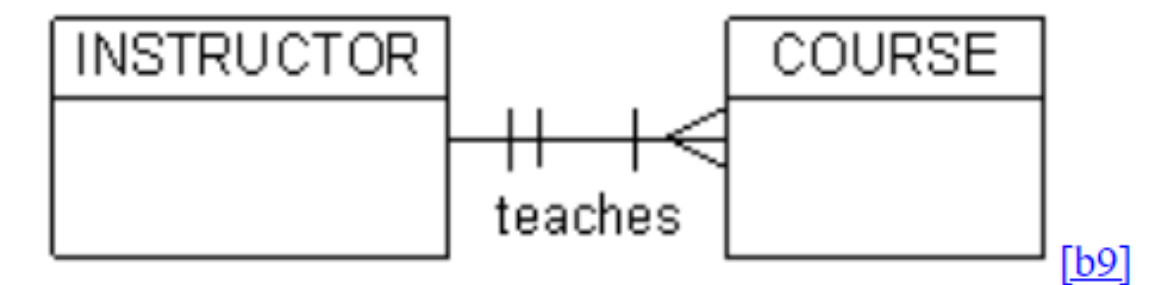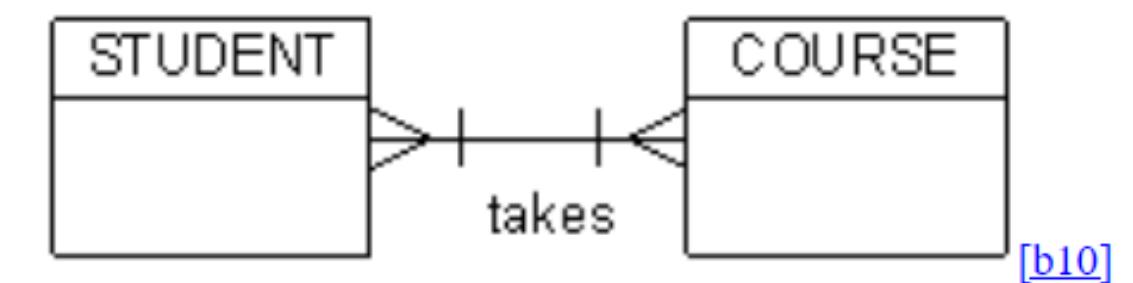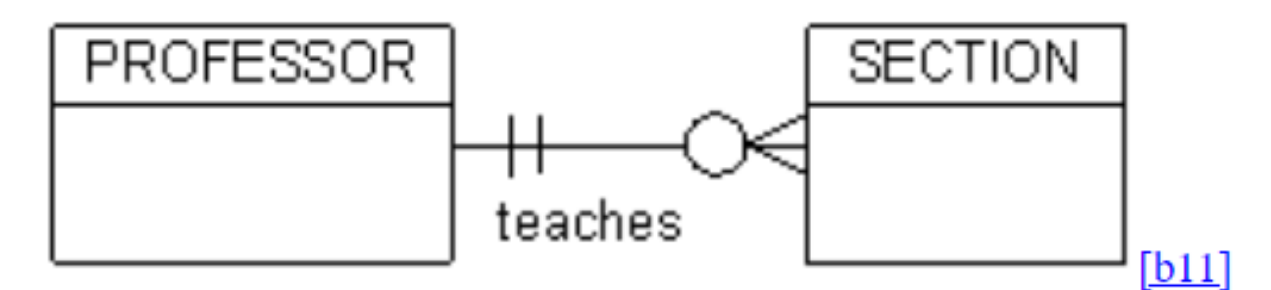
CONVERT ERD TO TABLES

- Each entity type becomes a table

- Each single-valued attribute becomes a column

- Derived attributes are ignored/computed column

- Multi-valued attributes are represented by a separate table

- Use Conjunction Table to break up the Many-to-Many relationship

- The key attribute of the entity type becomes the primary key or unique key of the table

CONVERT ERD TO TABLES
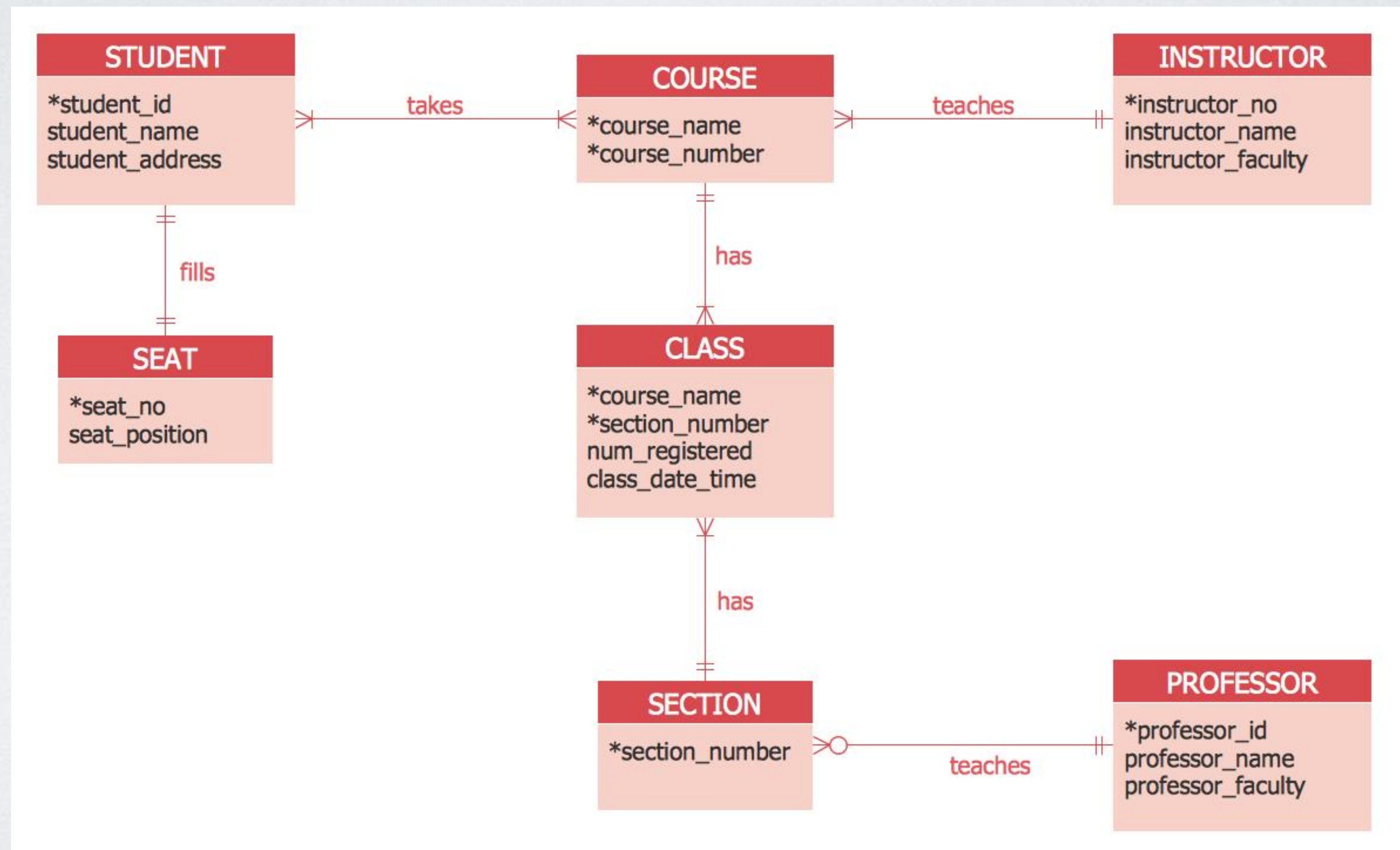
NORMALIZATION

- Redundancy

  - Values repeated unnecessarily in multiple records or fields within one or more tables

NORMALIZATION

- Anomaly

  - When an attempt is made to modify (update, insert into, or delete from) a relation, the following undesirable side-effects may arise in relations that have not been sufficiently normalized.

  - Anomaly is the issue that may occur because of redundancy

  - Types: Update, Insertion and Deletion

NORMALIZATION

- Anomaly Update

  - The same information can be expressed on multiple rows; therefore updates to the relation may result in logical inconsistencies.

NORMALIZATION

- Anomaly Insertion

  - There are circumstances in which certain facts cannot be recorded at all

Faculty and Their Courses

Faculty ID	Faculty Name	Faculty Hire Date	Course Code
389	Dr. Giddens	10-Feb-1985	ENG-206
407	Dr. Saperstein	19-Apr-1999	CMP-101
407	Dr. Saperstein	19-Apr-1999	CMP-201
424	Dr. Newsome	29-Mar-2007	?

NORMALIZATION

- Anomaly Deletion

  - Under certain circumstances, deletion of data representing certain facts necessitates deletion of data representing completely different facts

Faculty and Their Courses

Faculty ID	Faculty Name	Faculty Hire Date	Course Code
389	Dr. Giddens	10-Feb-1985	ENG-206
407	Dr. Saperstein	19-Apr-1999	CMP-101
407	Dr. Saperstein	19-Apr-1999	CMP-201

DELETE

NORMALIZATION

- Main goal of normalization

  - Reduce redundancy, avoid anomaly and create a well- structured series of table without error or inconsistencies.

  - Minimize redesign when extending the database structure

    - new type of data can be accommodated without changing existing structure too much

  - Ensures data dependencies are properly enforced by data integrity constraints (Entity Integrity, Referential Integrity, Domain Integrity)

NORMALIZATION

- Functional Dependencies

  - Functional Dependencies are how different attributes relate in a table.

  - At this level, we focus on individual tables

  - We see how individual attributes relate to the keys in the table

    - Primary Key & Candidate Keys = Prime Attributes

    - Attributes that aren't keys = Non-Prime Attributes

- Types of Dependencies

  - Full Dependencies -- Depends on all Prime Attributes Fully

  - Partial Dependencies -- Depends on some Prime Attributes

  - Transitive Dependences -- Depends on an attribute that depends on a Prime Attribute

1. The diagram below shows a relational schema and its functional dependencies. Label each dependency as a full, partial or transitive dependency. Convert the relation to a set of relations in second normal form. Then show the relations in 3rd normal form to illustrate the progression from a single relation to 2nd normal form and then 3rd normal form. For the set of relations in 3rd normal form, show the referential integrity constraints.

Project Number	Employee#	Project Name	Employee name	Job Class	Hourly Charge	Hours

1.

2.

3.

4.
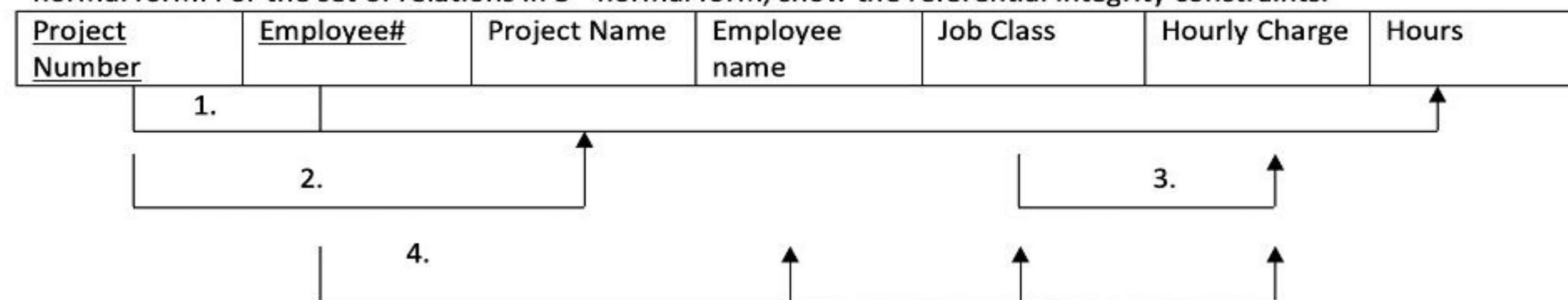
NORMALIZATION

- First Normal Form

  - Each table cell should contain a single value.

  - Each record needs to be unique.

- Second Normal Form

  - Meets all of 1NF

  - Makes sure all non-prime attributes are fully dependent on a prime attribute

- Third Normal Form

  - Meets 1NF and 2NF

  - Every non-prime attribute is non-transitively dependent on the prime attributes

PROCESS
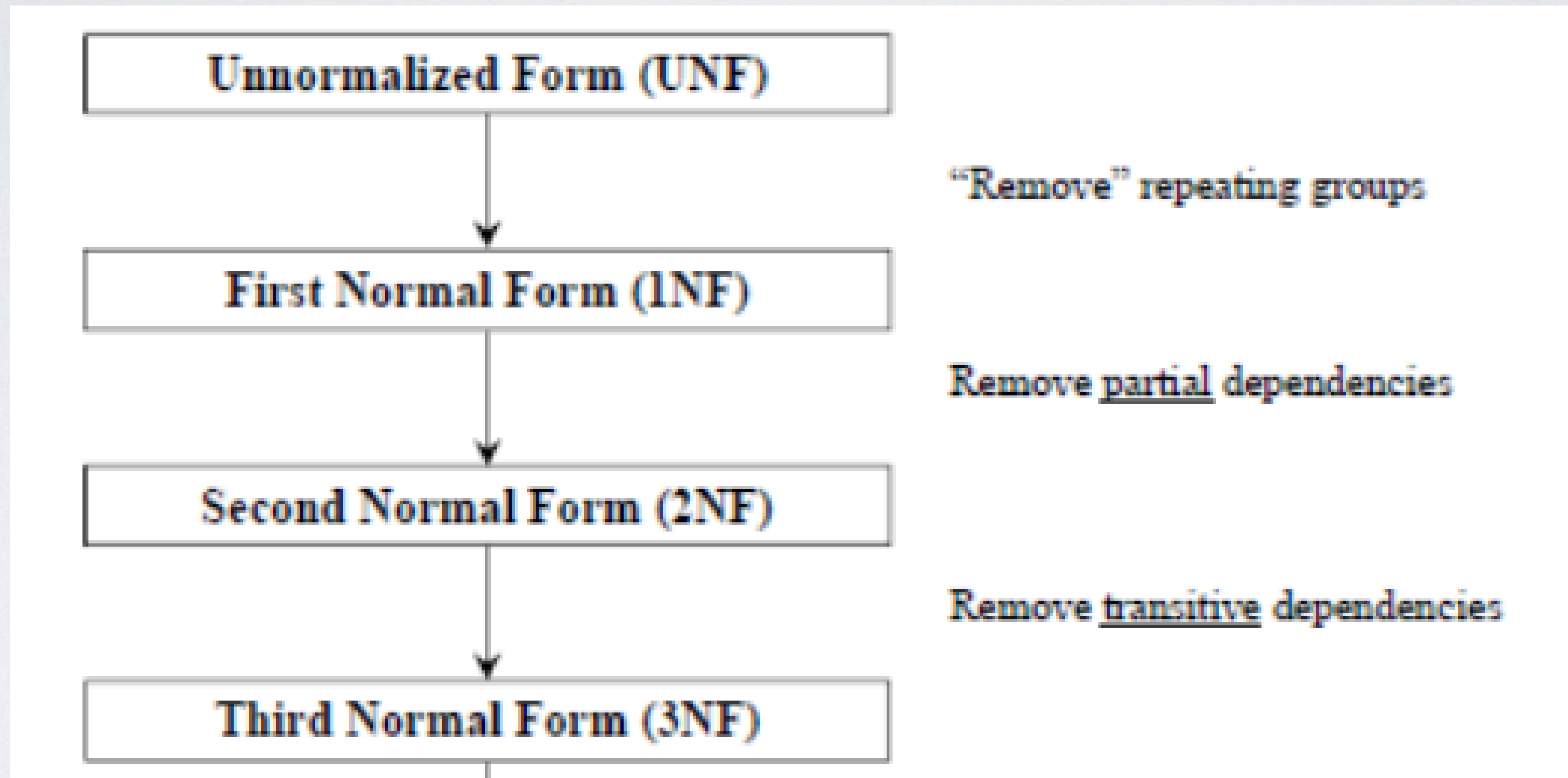
NORMALIZATION

- UNF

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.80
		101	John G. News	Database Designer	$105.00	19.40
		105	Alice K. Johnson	Database Designer	$105.00	35.70
		106	William Smithfield	Programmer	$35.75	12.60
		102	David H. Senior	Systems Analyst	$96.75	23.80
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.60
		118	James J. Frommer	General Support	$18.36	45.30
		104	Anne K. Ramoras	Systems Analyst	$96.75	32.40
		112	Darlene M. Smithson	DSS Analyst	$45.95	44.00
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.70
		104	Anne K. Ramoras	Systems Analyst	$96.75	48.40
		113	Delbert K. Joenbrood	Applications Designer	$48.10	23.60
		111	Geoff B. Wabash	Clerical Support	$26.87	22.00
		106	William Smithfield	Programmer	$35.75	12.80
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.60
		115	Travis B. Bawangi	Systems Analyst	$96.75	45.80
		101	John G. News	Database Designer	$105.00	56.30
		114	Annelise Jones	Applications Designer	$48.10	33.10
		108	Ralph B. Washington	Systems Analyst	$96.75	23.60
		118	James J. Frommer	General Support	$18.36	30.50
		112	Darlene M. Smithson	DSS Analyst	$45.95	41.40

NORMALIZATION

- After 1NF

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	$84.50	23.80
15	Evergreen	101	John G. News	Database Designer	$105.00	19.40
15	Evergreen	105	Alice K. Johnson	Database Designer	$105.00	35.70
15	Evergreen	106	William Smithfield	Programmer	$35.75	12.60
15	Evergreen	102	David H. Senior	Systems Analyst	$96.75	23.80
18	Amber Wave	114	Annelise Jones	Applications Designer	$48.10	24.60
18	Amber Wave	118	James J. Frommer	General Support	$18.36	45.30
18	Amber Wave	104	Anne K. Ramoras	Systems Analyst	$96.75	32.40
18	Amber Wave	112	Darlene M. Smithson	DSS Analyst	$45.95	44.00
22	Rolling Tide	105	Alice K. Johnson	Database Designer	$105.00	64.70
22	Rolling Tide	104	Anne K. Ramoras	Systems Analyst	$96.75	48.40
22	Rolling Tide	113	Delbert K. Joenbrood	Applications Designer	$48.10	23.60
22	Rolling Tide	111	Geoff B. Wabash	Clerical Support	$26.87	22.00
22	Rolling Tide	106	William Smithfield	Programmer	$35.75	12.80
25	Starflight	107	Maria D. Alonzo	Programmer	$35.75	24.60
25	Starflight	115	Travis B. Bawangi	Systems Analyst	$96.75	45.80
25	Starflight	101	John G. News	Database Designer	$105.00	56.30
25	Starflight	114	Annelise Jones	Applications Designer	$48.10	33.10
25	Starflight	108	Ralph B. Washington	Systems Analyst	$96.75	23.60
25	Starflight	118	James J. Frommer	General Support	$18.36	30.50
25	Starflight	112	Darlene M. Smithson	DSS Analyst	$45.95	41.40

DEPENDENCY
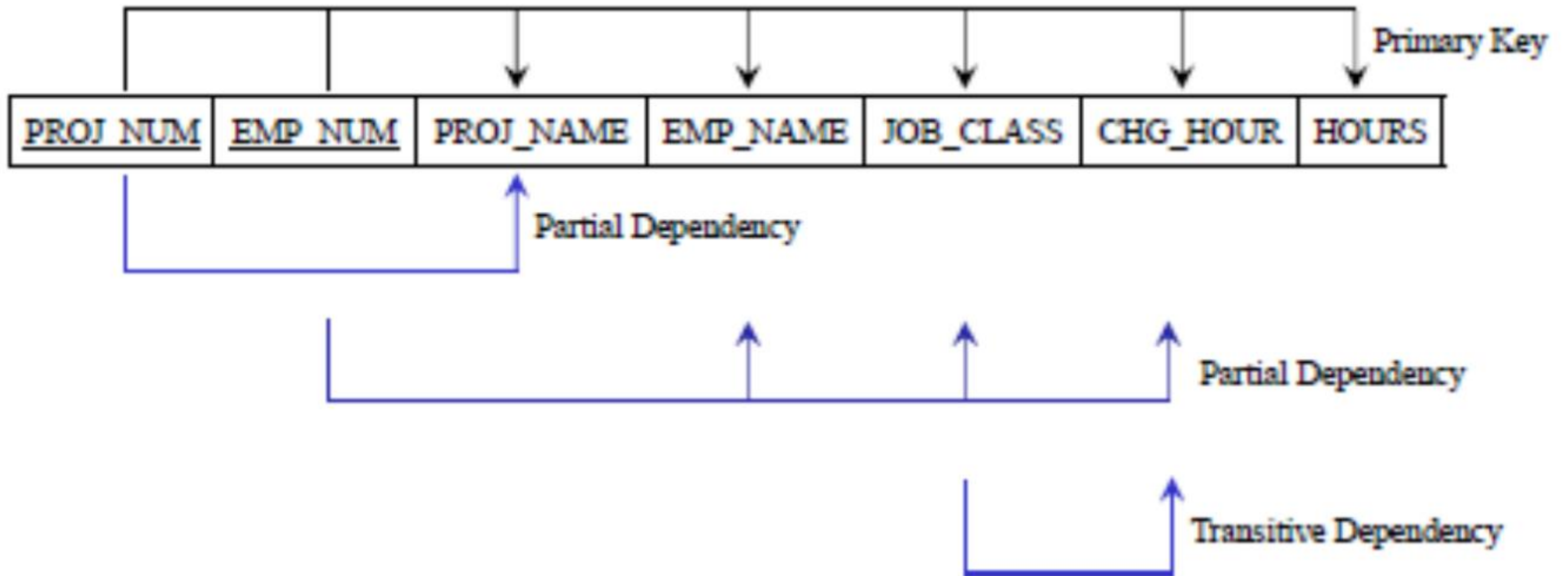
NORMALIZATION

- After 2NF
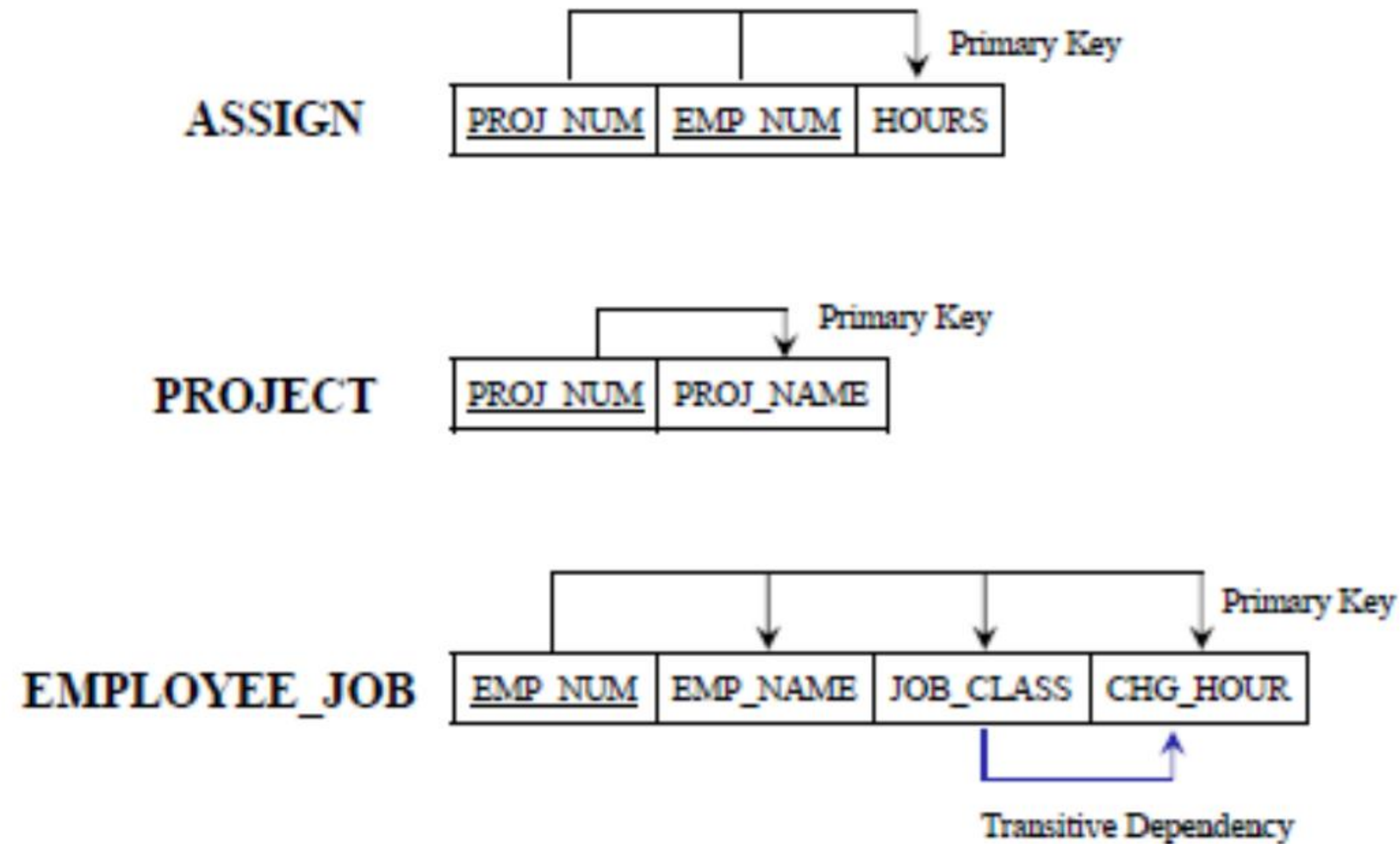
NORMALIZATION

- 3 NF (No transitive dependency)
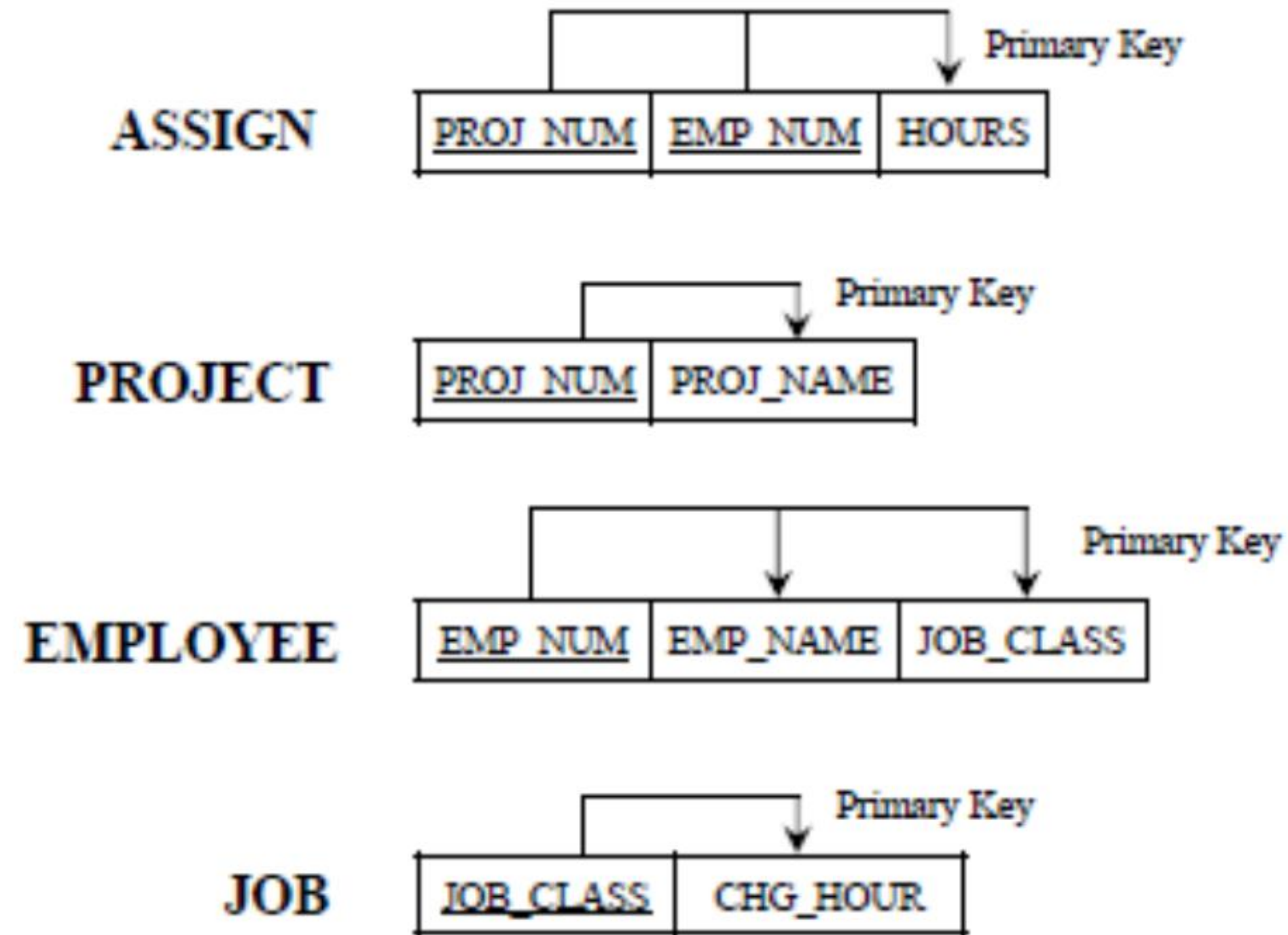
NORMALIZATION

- When Do We Normalize?

  - Usually normalize in database

  - Lots of read/writes

  - Not too much fetching/only need to fetch a small subset of the data

  - Data integrity is crucial especially because there is a lot of user input

- Cons

  - Normalization is an expensive process

  - Designing can be difficult – Good for final designs, not testing

  - More tables leads to more time

  - Joins are costly, having more tables can cause slowing

DATABASE INTEGRITIES

- Entity Integrity

  - Design of the table or entity

  - Strong PK with no nulls or repeats

- User-Defined Integrity

  - Rules or constraints applied by the user to maintain rules of design

- Domain Integrity

  - Correct and proper domains specified with proper use of columns

- Referential Integrity

  - Proper FK setup with proper PK reference

  - Good design for connection and joins

DATABASE MANAGEMENT SYSTEM

RELATIONAL DATABASE VS. NON-RELATIONAL DATABASE

- Relational Database

  - Traditional way of storing data

  - Based on relational model of data

  - Contains tables and relations

  - Uses SQL language

  - Difficult to scale

- Non-relational Database

  - Developed more recently

  - For some of the data cannot be stored or is difficult to store in relational database

  - NoSQL

  - Can take various forms

  - Modeling method is not well established

  - Easily scalable and appendable

- Database Ranking: https://db-engines.com/en/ranking

SQL (STRUCTURED QUERY LANGUAGE)

- DDL – Data Definition Language

  - Create

  - Alter

  - Drop

  - Truncate

```
1  ⊟CREATE TABLE TEST1
2   (
3     COLUMN1 Int
4    ,COLUMN2 Varchar(10)
5    ,COLUMN3 VarChar(10)
6   );
7
8  ⊟ALTER TABLE TEST1
9   ALTER COLUMN COLUMN3 CHAR(10)
10
11  DROP TABLE TEST1
12
13
```

Create

This command builds a new table and has a predefined syntax.

The CREATE statement syntax is:

CREATE TABLE [table name] ([column definitions]) [table parameters];

For example:

CREATE TABLE Employee (Employee Id INTEGER PRIMARY KEY, First name CHAR (50) NULL, Last name CHAR (75) NOT NULL);

The mandatory semicolon at the end of the statement is used to process every command before it. In this example, the string CHAR is used to specify the data type. Other data types can be DATE, NUMBER, or INTEGER.

Alter

An alter command modifies an existing database table. This command can add up additional column, drop existing columns and even change the data type of columns involved in a database table.

An alter command syntax is:

ALTER object type object name parameters;

For example:

ALTER TABLE Employee ADD PRIMARY KEY (employee_pk);

In this example, we added a unique primary key to the table to add a constraint and enforce a unique value. The constraint "employee_pk" is a primary key and is on the Employee table.

Drop

A drop command is used to delete objects such as a table, index or view. A DROP statement cannot be rolled back, so once an object is destroyed, there's no way to recover it.

Drop statement syntax is:

DROP object type object name;

For example:

DROP TABLE Employee;

In this example, we're deleting the Employee table.

Truncate

Similar to DROP, the TRUNCATE statement is used to quickly remove all records from a table. However, unlike DROP that completely destroys a table, TRUNCATE preserves its full structure to be reused later.

Truncate statement syntax is:

TRUNCATE TABLE table_name;

For example:

TRUNCATE TABLE Employee;

In this example, we're marking all the extents of the Employee table for deallocation, so they're considered empty for reuse.

SQL

- Adding constraints to a table

  - Constraints: used to specify rules for the data in table.

  - Type: Not null, Null, Unique, Primary key, foreign key, check, default.

  - How?

    - Create table then alter to add constraints

    - Add constraint as we specify the column in a table

    - Add constraint in the same create statement, after we specified the table

SQL

- Create table then alter to add constraints

```
ALTER TABLE edw.API
    ADD CONSTRAINT PK__API PRIMARY KEY CLUSTERED (API_Id),

    CONSTRAINT FK__API_Type_Id_API
        FOREIGN KEY (API_Type_Id) REFERENCES edw.API_Type(API_Type_Id),

    CONSTRAINT DF__API__Create_Dt default GETDATE() for Create_Dt,

    CONSTRAINT DF__API__Create_User_Id default SYSTEM_USER for Create_User_Id,

    CONSTRAINT DF__API__Update_Dt default GETDATE() for Update_Dt,

    CONSTRAINT DF__API__Update_User_Id default SYSTEM_USER for Update_User_Id;
```

- Add constraint as we specify the column in a table

```
CREATE TABLE #FUNDOVERNIGHT_ACCT_IDLIST
(
    ACCT_ID Char(12) PRIMARY KEY
);
```

- Add constraint in the same create statement, after we specified the table

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    CONSTRAINT PK_Person PRIMARY KEY (ID,LastName)
);
```

SQL

- DML-Data Manipulation Language

  - Insert

  - Update

  - Delete

DML

- Insert Statements

  - Insert statements are used to input data into tables

  - The order of data specified should match order of columns

  - If you don't know the order of columns, specify each column name in the insert statement

  - For example

    - Inserting data matching order

      - Insert into TblName values(1,'Name'),(2,'Name);

    - Inserting data without knowing matching order

      - Insert into TblName (Col2,Col1) values('Name',1),('Name',2)

      - the order of col matter?

DML

- Update Statements

  - Update statements are used to change or modify data inside a table

  - Specify single row depending on statement

  - Use unique identifiers to get correct rows

  - For Example

    - Update using unique column

      - Update TblName Set Name = 'Tim' Where ID = 2

    - Update using non-unique column

      - Update TblName Set Name = 'Hal' Where Name = 'Bob'

      - How about without using WHERE clause?

DML

- Delete Statements

  - Delete statements are used to remove specific rows inside a table.

  - Use unique values to identify the correct rows to delete

  - Deletes leave logs and continue identity values

  - For Example

    - Deleting rows using specific unique values

      - Delete From TblName Where ID = 1

    - Deleting rows using non-unique values

      - Delete From TblName Where Name = 'Jim'

    - without WHERE clause?

SQL

- DCL-Data Control Language

    - DCL is syntax used to control what permission a user can have.

    - You can create Roles that users can be grouped into a specific permission there

    - You can choose what tables someone in a role can access and even what statements can be performed

DCL

- Grant

  - Used to "grant" or give permissions to a user or role

  - Examples

    - Create table permission to a role

      - Grant Create Table to TestRole

    - Add user to the role

      - Exec sp_addrolemember 'TestRole', 'TestUser'

    - Search "mySQL grant" to check out all the privilege

    - https://dev.mysql.com/doc/refman/8.0/en/grant.html

```
GRANT [privilege]
ON [object]
TO [user]
[WITH GRANT OPTION]
```

```
GRANT SELECT
ON HR.employees
TO Joe
```

DCL

- Revoke

  - Revoke is used to take away permissions given, whether they are Grant or Deny permissions.

  - Examples

    - Revoking grant permission

      - Revoke Create Table to TestRole

```
REVOKE [GRANT OPTION FOR] [permission]
ON [object]
FROM [user]
[CASCADE]
```

```
REVOKE SELECT
ON HR.employees
FROM Joe
```

DCL

- Deny

  - Deny is preventing someone from having access at all.

  - Deny is not the same as revoke, as revoke is made to take back, deny is made to say NO

  - If grant and deny permission are given to the same role, deny will take over

  - Example

    - Deny Create Table to TestRole

```
DENY [permission]

ON [object]

TO [user]
```

```
DENY DELETE

ON HR.employees

TO Matthew
```

SQL

- DQL –Data Query Language

  - Select From Where

    - Select: Pick up which column of data you'd like to fetch

    - From: Select which table or data set to fetch.

    - Where: Specific a criteria to sort data by use operators. (Filter)

      - Operators: In, Or, And

  - Group by, Having, Order by

    - Group by – Used to combine similar values in column

    - Having – filter conditions for aggregate only

    - Order by – display the data by order by a specific column

```
SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
HAVING COUNT(CustomerID) > 5
ORDER BY COUNT(CustomerID) DESC;
```

CONSTRAINT

- Constraints

  - A constraint is usually associated with a table and is created with a CREATE CONSTRAINT SQL statement.

  - They define certain properties that data in a database must comply with.

- Key Constraints

  - Primary Key

    - 1 per table

    - Unique Clustered Index

    - Not Null

  - Unique Key

    - 999 Per Table

    - Unique NonClustered Index

    - 1 Null Allowed

  - Foreign Key

    - Can not exist before PK

    - Must be deleted before PK

- Other Constraints

  - Null, Not null

    - Are nulls allowed

  - Check

    - Data must meet rule

  - Default

    - If nothing then this

  - Data types (Kind of)

    - Char (2) – States

    - Varchar (12) – Phone

    - Money – Money!

ANY QUESTIONS?