RDBMS

Day 1

Agenda

- Overview of Database Management and Architecture
- Introducing Relational Databases
- Normalization
- Retrieving Data Using SQL

RDBMS

Module 1: Overview of Database Management and Architecture

Database management system

- A computer program (or more typically, a number of them) designed to manage a database, a large set of structured data, and run operations on the data requested by a number of users is called Database Management System (DBMS).
- A few examples of DBMS use include accounting, human resources and customer support systems.

Entities in a database

- Data.
- Container of data.
- Application Program.

These three entities are independent.

Why database

The three primary advantages that we get when using a database over a file system, are

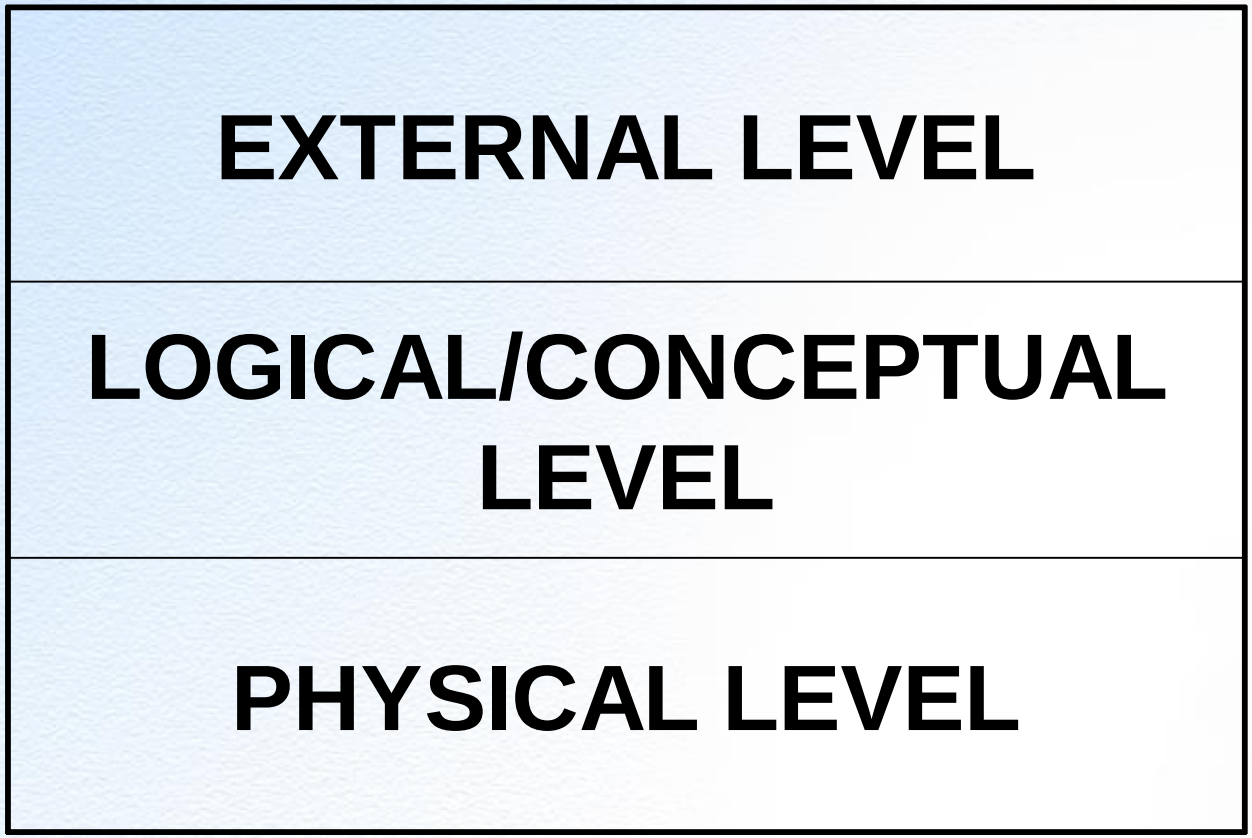
- Program – Data Dependency.
- Rigidity.
- Data Redundancy and Inconsistency.

Components of database system

Four components of database system are:

- Data.
- Software.
- Hardware.
- Users.

Layered database architecture



Features of database system

- Sharing of data.
- Reduction of Redundancy.
- Avoidance of Inconsistency.
- Maintenance of Integrity.
- Enforcement of Security and Standards.
- Transaction Support can be provided.
- Conflicting requirements can be balanced.

Database users

- End Users.
- Application Programmers.
- Database Administrators.

Mapping: External conceptual mapping

- The logical level contains the representation of data, indirectly data would be stored in files at the physical level. External conceptual mapping is a mapping between a view at the external level with an object at the logical level.

Conceptual internal mapping

- The mapping of objects at the logical level of the database with the files at the physical level.

Data independence

- The separation of data from the programs that use the data.
- Nearly all modern applications are based on the principle of data independence. In fact, the whole concept of a database management system (DBMS) supports the notion of data independence since it represents a system for managing data separately from the programs that use the data. In contrast, it is possible to write applications in which the data being processed is actually represented in the program's source code.
- This data-dependent approach is very inflexible because it makes it difficult to modify the data and it also makes the data inaccessible to other programs.
- It is the immunity of applications to change in physical representation and access technique.

Physical data independence

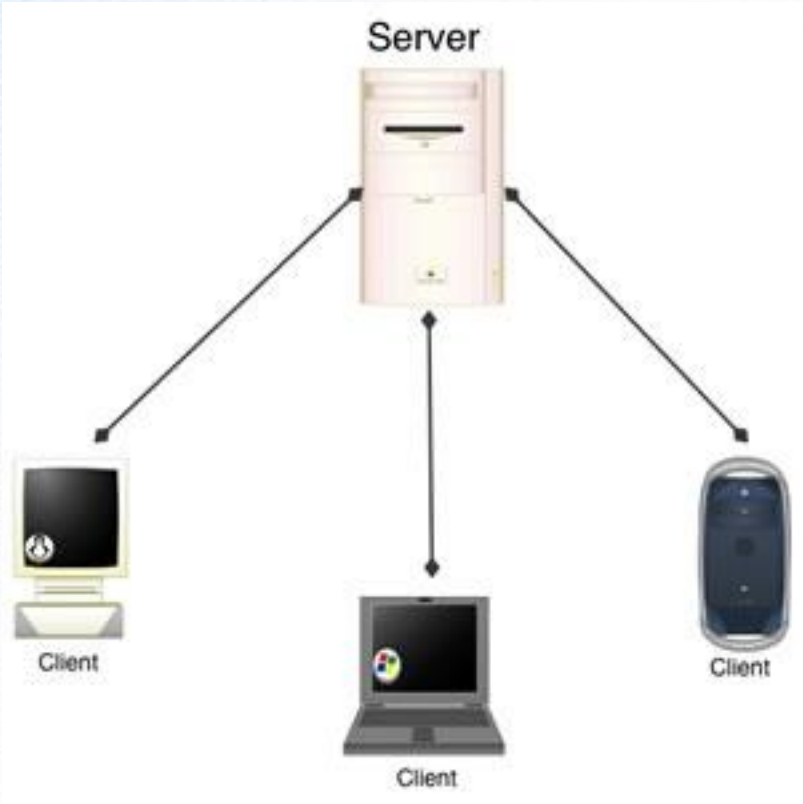
- It is the capacity to change the internal schema of a database without having to change the conceptual schema.
- All databases guarantee physical data independence.
- Affecting physical files, in no way should affect the logical objects.

Logical data independence

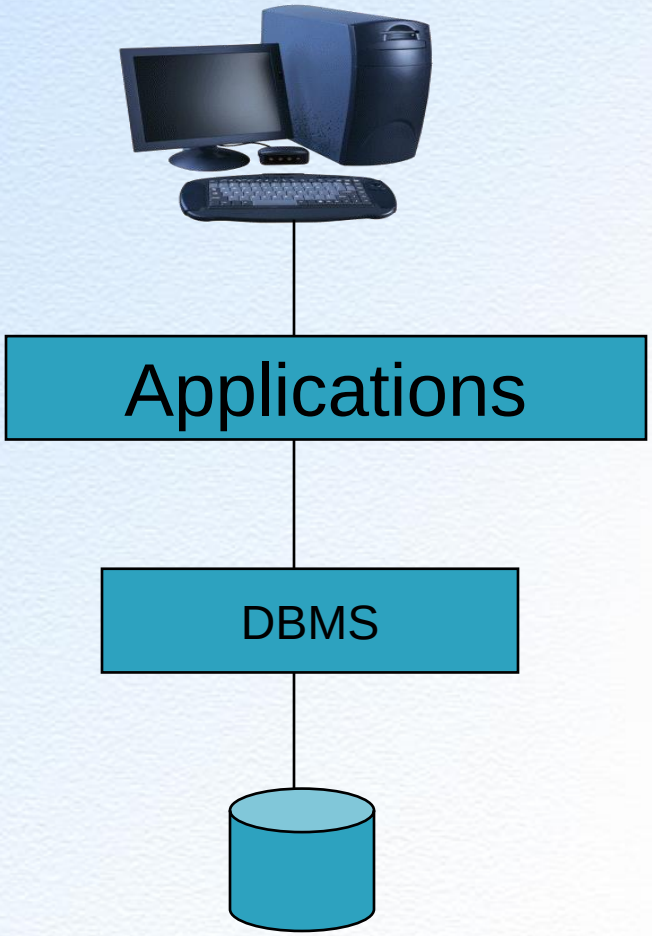
- It can be defined as the capacity to change the conceptual schema without having to change the external schema or application programs.
- What happens if changes are made at the logical level of the database? Eg: Alter Table, Rename Table etc. It is brought about by setting restrictions.
- The more restrictive the database, the more logical data independence will one have.

Client/server architecture

- **Server:** It is basically the DBMS itself. It supports all the basic DBMS functions.
- **Clients:** They are various applications run on top of DBMS.
  - User-written applications.
  - Vendor-provided applications



Client/server architecture (continued)



End Users

Clients

Server

Database

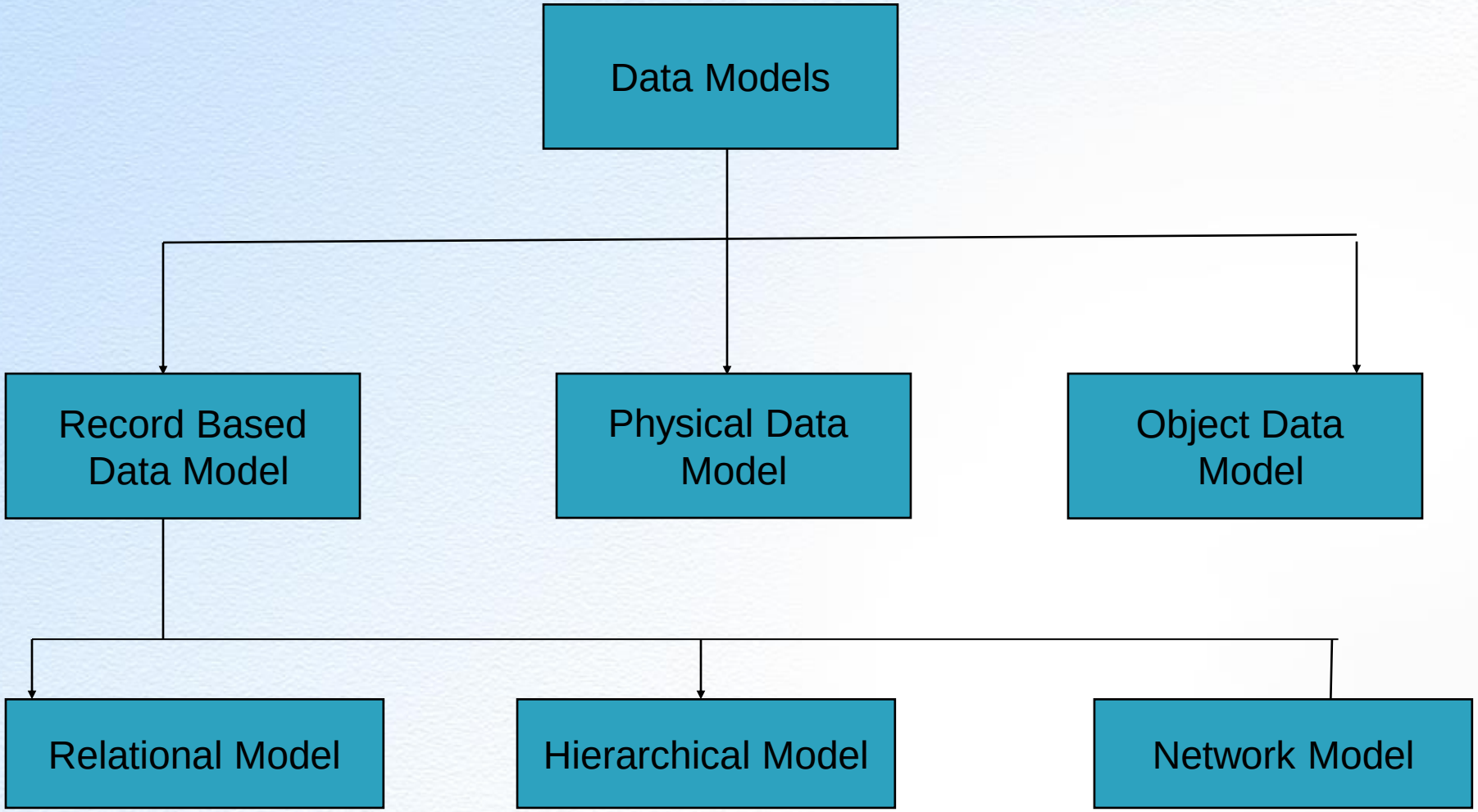
RDBMS

Module 2: Introducing Relational Databases

Data models

- Giving a description to the logical or conceptual level.
- It is the representation of data in a database at the logical level.

Data models (continued)

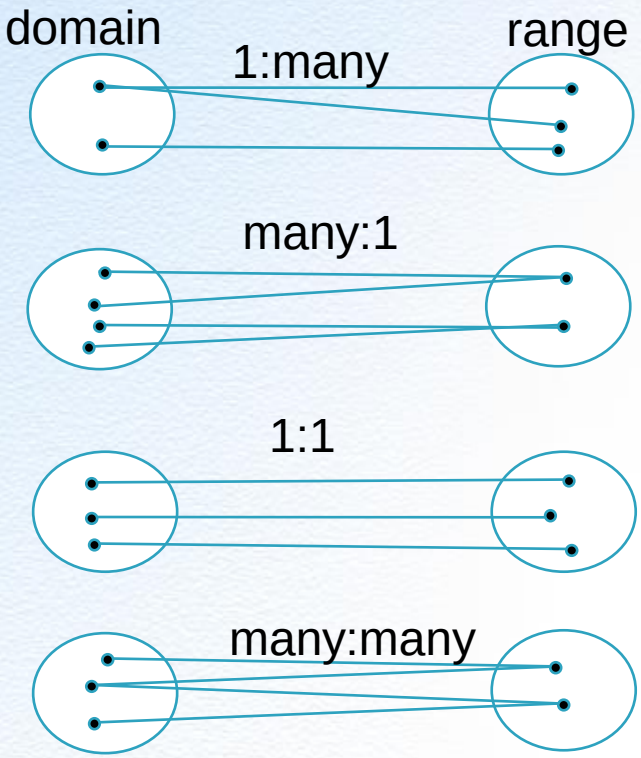


Relational database

The important features are:

- Name of the object at the logical level.
- Structure of the logical level object.

Relationship mapping



Relationship model

- No duplicate tuples should be there in a relation.
- Ordering of tuples is not necessary.
- Attributes of a relation have an implied ordering
- The attribute values of a tuple must be within the domain of the relation or a special value *NULL must be present.*
- The attribute values and domain values must be atomic.

Comparative Terms

Terms	Oracle
Relation schema	Table description
Relation	Table
Tuple	Row
Attribute	Column
Domain	Value set

- **Notation**

Employee (emp_id,emp_name,sal)
Student (studno,name,class)
Department (Dept_no,Dept_name)

Keys

- Candidate Key.
- Primary Key.
- Super key.
- Foreign Key.
- Alternate Key.

Key definitions

- **Candidate Key:** There may be a number of attributes or a combination of attributes which uniquely identify each instance of an entity. These are known as candidate keys.
- **Primary Key:** An attribute whose value can uniquely identify a complete record (one row of data) within an entity.
- **Super key:** If additional attributes are added to the primary key, the resulting combination would still uniquely identify an instance of the entity set. These augmented keys are called super keys.
- **Composite Primary Key:** A primary key that consists of two or more attribute within an entity.

Key definitions (continued)

- **Foreign Key:** A copy of a primary key that exists in another entity (may exist in the same entity too) for the purpose of forming a relationship between the entities involved. (Discussed in detail later.)
- **Alternate Key:** The primary key is selected from the set of candidate keys. The remaining candidate keys are the alternate keys.

Foreign key

- a set of attributes in a relation that exactly matches a (primary) key in another relation
  - the names of the attributes don't have to be the same but must be of the same domain
  - a foreign key in a relation A matching a primary key in a relation B represents a
- many: one relationship between A and B
- Employee (Emp_id, Emp_name, Sal, Dept_id)
- Department (Dept_id, Dept_name)

Referential integrity

- Cascade: delete all matching foreign key tuples. Deletes those tuples matching in employee table.
- Restrict: can't delete primary key tuple Department whilst a foreign key tuple Employee matches
- Nullify: foreign key Employee.Dept_id set to null if the primary key Dept_ids are deleted.

Entity integrity and nulls

No part of a key can be null

- Attribute values
  - Atomic
  - Known domain
  - Sometimes can be null

THREE categories of null values

- Not applicable
- Not known
- Absent (not recorded)

Attribute constraints

- This constraint is basically the declaration that a specified attribute is of specified type, the maximum value, minimum value, length of the field associated with them.
- Example:

```
Emp Relation
 { Emp_no Integer(5)
    Emp_name Char(20)
 }
```

Database constraints (Oracle)

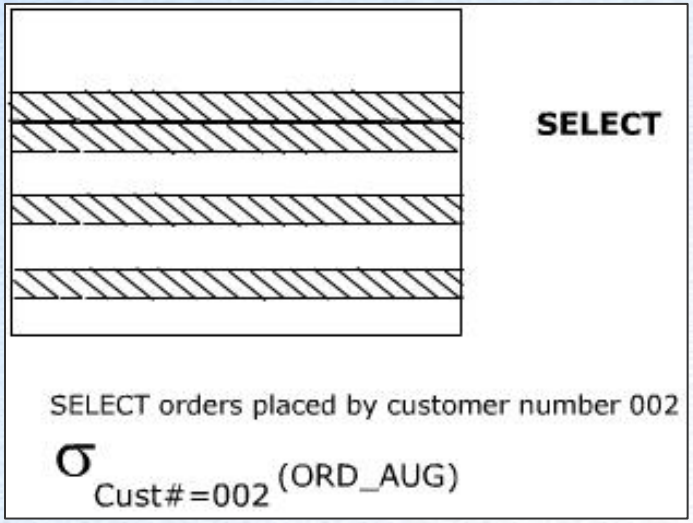
- Not Null.
- Check.
- Primary Key.
- Foreign Key.
- Unique.

Relational Operators

- **Restrict orSelect:** Only specific tuples from a relation satisfying a specific condition is returned.
- **Project:** All tuples in a specified relation and only specified attributes are returned.
- **Product:** It returns a relation containing all possible tuples that are a combination of two tuples, one from each of two specified relations.
- **Union:** All tuples appearing in either or both of two specified relations are returned.
- **Intersect:** All tuples appearing in both of two specified relations are returned.
- **Difference:** All tuples appearing in the first and not the second of two specified relations are returned.
- **Join:** All tuples that are a combination of two tuples, one from each of two specified relations, such that the two tuples contributing to any given combination have a common value for the common attributes of the two relations.

Select

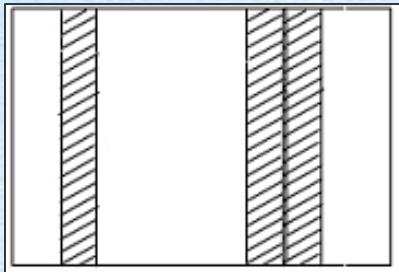
- To retrieve specific tuples/rows from a relation



Ord#	OrdDate	Cust#
101	02-08-94	002
104	18-09-94	002

Project

- To retrieve specific attributes/columns from a relation.



Project

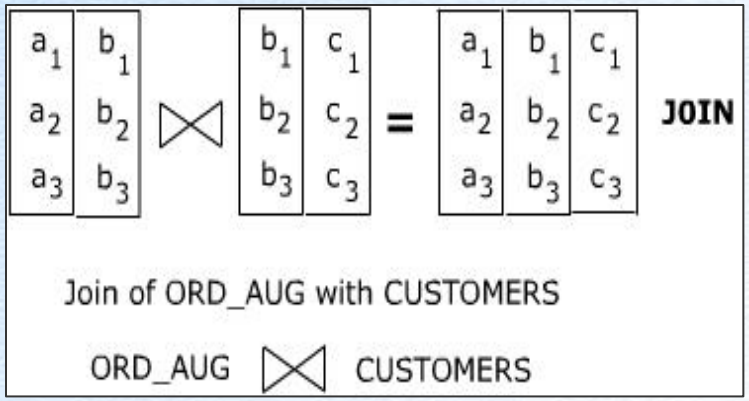
PROJECT description and price from Items relation

$$\Pi_{\text{Descr, Price}}(\text{ITEMS})$$

Descr	Price
Power Supply	4000
101-Keyboard	2000
Mouse	800
MS-DOS 6.0	5000
MS-Word 6.0	8000

Join

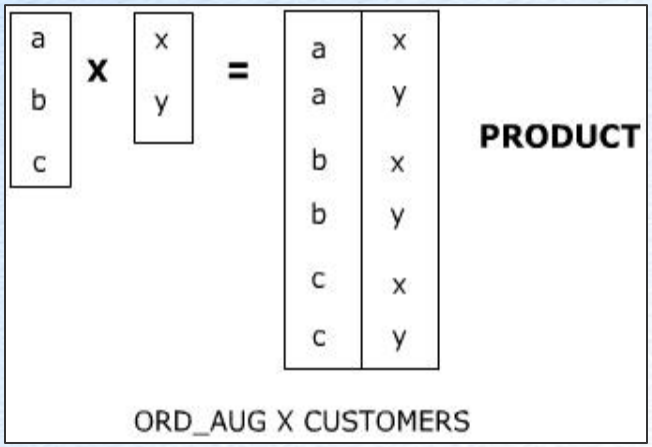
- Example:
  ORD_AUG join CUSTOMERS (here, the common column is Cust#)



Ord#	OrdDate	Cust #	CustNames	City
101	02-08-94	002	Srinivasan	Madras
102	11-08-94	003	Gupta	Delhi
103	21-08-94	003	Gupta	Delhi
104	28-08-94	002	Srinivasan	Madras
105	30-08-94	005	Apte	Bombay

Product

- To obtain all possible combination of tuples from two relations.



Ord#	OrdDate	O.Cust#	C.Cust#	CustName	City
101	02-08-94	002	001	Shah	Bombay
101	02-08-94	002	002	Srinivasan	Madras
101	02-08-94	002	003	Gupta	Delhi
101	02-08-94	002	004	Banerjee	Calcutta
101	02-08-94	002	005	Apte	Bombay
102	11-08-94	003	001	Shah	Bombay
102	11-08-94	003	002	Srinivasan	Madras

Tables

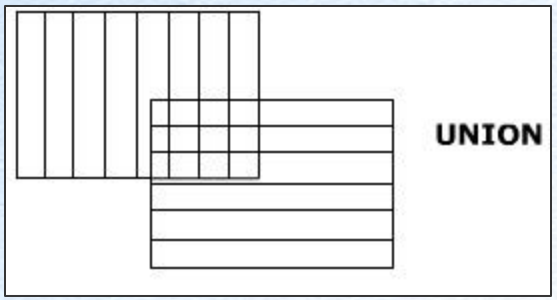
- These are the reference tables on which the next operations will be performed.

Emp_no	E_Name
1	Jones
3	Smith
4	Lalonde
7	Evan
10	Drew
12	Smith

Emp_no	E_Name
3	Smith
4	Lalonde
6	Byron
10	Drew

Union

- To retrieve tuples appearing in either or both the relations participating in the UNION.

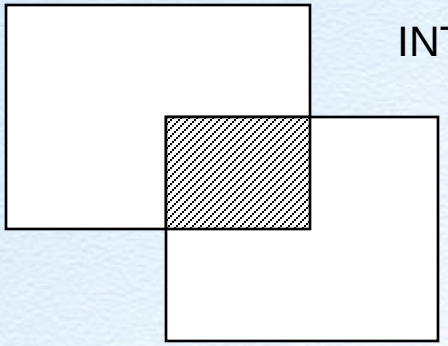


Emp_no	E_name
1	Jones
3	Smith
4	Lalonde
6	Byron
7	Evan
10	Drew
12	Smith

AUB

Intersect

- To retrieve tuples present in both the relations participating in the intersect.



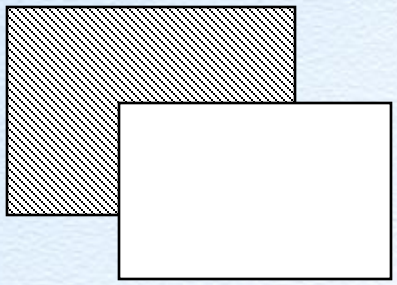
INTERSECT

Emp_no	E_name
3	Smith
4	Lalonde
10	Drew

$A \cap B$

Difference

- To retrieve tuples appearing in the first relation participating in the DIFFERENCE but not the second.



DIFFERENCE

Emp_no	E_name
1	Jones
7	Evan
12	Smith

A-B

Exercise

- The remaining candidate keys left out after the selection of the primary key are called _____.
- The operator used to retrieve specific attributes from a relation is called the _____ operator.
- The constraint which disallows the occurrence of duplicate values for an attribute is known as the _____ constraint.
- The primary key – foreign key relationship sets _____ integrity among two relations.
- When a primary key is formed from multiple attributes in a relation, it is called a _____ key.

RDBMS

Module 3: Normalization

The purpose of normalization

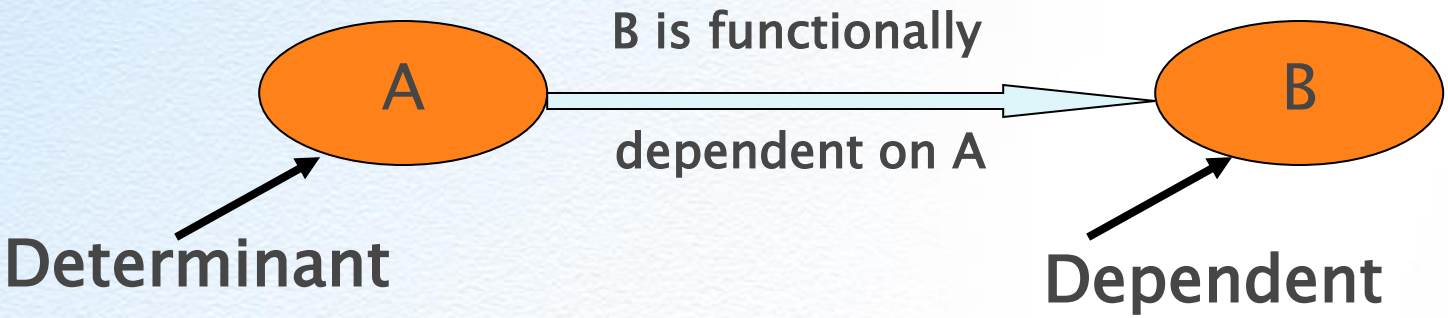
- **Normalization** is a technique for producing a set of relations with desirable properties, given the data requirements of an enterprise.

The process of normalization

- Normalization is often executed as a series of steps. Each step corresponds to a specific normal form that has known properties.
- As normalization proceeds, the relations become progressively more restricted in format, and also less vulnerable to update anomalies.
- For the relational data model, it is important to recognize that it is only first normal form (1NF) that is critical in creating relations. All the subsequent normal forms are optional.

Functional dependencies

- Functional dependency describes the relationship between attributes in a relation.
- For example, if A and B are attributes of relation R, and B is functionally dependent on A(A:B), if each value of A is associated with exactly one value of B. (A and B may each consist of one or more attributes.)



Example: Functional dependencies

S#	CITY	P#	QTY
S1	DELHI	P1	100
S1	DELHI	P2	100
S2	KOLKATA	P1	200
S3	KOLKATA	P2	300
S4	DELHI	P2	400
S4	DELHI	P4	400
S4	DELHI	P5	400

{S#} : {CITY}

Non-loss decomposition

SUP

S#	STATUS	CITY
S1	10	KOLKATA
S2	10	MUMBAI

SST

S#	STATUS
S1	10
S2	10

SC

S#	CITY
S1	KOLKATA
S2	MUMBAI

SST

S#	STATUS
S1	10
S2	10

STC

STATUS	CITY
10	KOLKATA
10	MUMBAI

Supplier_details

S#	CITY	STATUS	P#	QTY
S1	DELHI	20	P1	300
			P2	200
			P3	400
			P4	200
			P5	100
			P6	100
S2	KOLKATA	10	P1	300
			P2	400
S3	KOLKATA	10	P2	200
S4	DELHI	20	P2	200
			P4	300
			P5	400

First normal form (1NF)

- A relation is in 1nf is all its underlying domains contain only scalar values. The above relation is in 1NF.

Table supplier_details (1NF)

S#	CITY	STATUS	P#	QTY
S1	DELHI	20	P1	300
S1	DELHI	20	P2	200
S1	DELHI	20	P3	400
S1	DELHI	20	P4	200
S1	DELHI	20	P5	100
S1	DELHI	20	P6	100
S2	KOLKATA	10	P1	300
S2	KOLKATA	10	P2	400
S3	KOLKATA	10	P2	200
S4	DELHI	20	P2	200
S4	DELHI	20	P4	300
S4	DELHI	20	P5	400

First normal form (1NF) (continied)

- INSERT Anomaly.
- UPDATE Anomaly.
- DELETE Anomaly.

In the Supplier_Details relation we find that the non-key attribute CITY is functionally dependent on the attribute S#, i.e part of the primary key.

Second normal form (2NF)

- A relation is in 2NF if and only if it is in 1NF and every non-key attribute is irreducibly dependent on the primary key.

Table supplier_details (2NF)

S#	STATUS	CITY
S1	20	DELHI
S2	10	KOLKATA
S3	10	KOLKATA
S4	20	DELHI
S5	30	MUMBAI

Supplier

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

Second normal form (2NF) (continued)

- INSERT Anomaly.
- DELETE Anomaly.
- UPDATE Anomaly.

In the Supplier relation we find that the non-key attributes STATUS and CITY are dependent on each other. Hence there is transitive dependency.

Third normal form (3NF)

Transitive dependency

- A condition where A, B, and C are attributes of a relation such that if A ⋅ B and B ⋅ C, then C is transitively dependent on A via B (provided that A is not functionally dependent on B or C).
- The normalization of 2NF relations to 3NF involves the removal of transitive dependencies by placing the attribute(s) in a new relation along with a copy of the determinant.

Third normal form (3NF)

- A relation is in 3NF if and only if it is in 2NF and every non-key attribute is non-transitively dependent on the primary key.

Table supplier_details (3NF)

S#	CITY
S1	DELHI
S2	KOLKATA
S3	KOLKATA
S4	DELHI
S5	MUMBAI

CITY	STATUS
DELHI	20
KOLKATA	10
MUMBAI	30
CHENNAI	40

SupCity

Denormalization

- The technique of taking normalized relations and making them non-normalized thus in the process bringing about redundancy is called denormalization.

Introduction to Oracle

Objectives

- Identify the features of Oracle
- Understanding the importance of SQL
- Background of SQL

Features of Oracle

- Oracle is an internet based object RDBMS.
- It consists of different relations, relational operations, indexes and constraints imposed on the relations.
- Oracle Server supports the SQL language and PL/SQL engine for storage and management of information.
- Oracle is scalable and reliable.

Uses of SQL

- SQL is Structured Query Language. It is used for accessing and manipulating database systems.
- It is used to perform queries on tables and views.
- It is also used to perform the following operations on tables.
- Data Manipulation Language (DML) operations:
  - INSERT
  - UPDATE
  - DELETE

Background of SQL

- SQL was developed by the IBM Corporation in the late 70's and was endorsed as a national standard by the American National Standard Institute (ANSI) in 1992. Sql 3 a version of SQL incorporates some object-oriented concepts.
- SQL is not a complete programming language, instead it is called a data sublanguage. SQL statements can be used in different ways. They can be used simply for DBMS processing and they can also be used as embedded SQL statements into client/server application programs.

Categories of SQL statements

- Data Retrieval Language(DRL)
- Data Manipulation Language (DML)
- Data Definition Language (DDL)
- Transaction Control Language (TCL)
- Data Control Language (DCL)

RDBMS

Module 4: Retrieving Data Using SQL

SELECT statement

- Projection
- Selection
- Join

Basic select statement

Select ?

From ?

- Select identifies what columns
- From identifies which table

Basic select statement (continued)

- Select * from employees;
- Select employee_id,salary from employees;

Writing SQL statements

- SQL statements are not case sensitive.
- SQL statements can be on one or more lines.
- Keywords cannot be abbreviated or split across lines.
- Clauses are usually placed on separate lines.
- Indents are used to enhance readability.

Column heading defaults

- *i*SQL*Plus:
  - Default heading justification: Center
  - Default heading display: Uppercase
- SQL*Plus:
  - Character and Date column headings are left-justified
  - Number column headings are right-justified
  - Default heading display: Uppercase

Arithmetic expressions

- Arithmetic Expression can have date or number
- Arithmetic Operators

  +,-,*,/

Using arithmetic operators

- Select first_name,salary,salary*12 from employees;
- Select first_name,salary+1000 from employees;

Operator precedence

- Multiplication and division take priority over addition and subtraction.
- Operators of the same priority are evaluated from left to right.
- Parentheses are used to force prioritized evaluation and to clarify statements.

Operator precedence (continued)

- Select first_name,salary,12*salary+1000 from employees.
- Select first_name,salary,12*(salary+1000) from employees.

NULL value

- A null is a value that is unavailable, unassigned, unknown, or inapplicable.
- A null is not the same as zero or a blank space.
- select first_name, salary, commission_pct from employees;

Defining a Column Alias

A column alias:

- Renames a column heading
- Is useful with calculations
- Immediately follows the column name – there can also be the optional AS keyword between the column name and alias
- Requires double quotation marks if it contains spaces or special characters or is case sensitive

Using column aliases

- Select first_name as name, job_id job employees;
- Select first_name “Name”,salary*12 “Annual Salary” from employees

Concatenation Operator

A concatenation operator:

- Concatenates columns or character strings
- Represented by two vertical bars (||)
- Creates a resultant column that is a character expression

Using the concatenation operator

- Select first_name||last_name “Full Name“ from employees;
- Select last_name|| ‘earns ‘||salary from employees;

Displaying distinct rows

- Select distinct salary from employees;

Using where clause

- Selecting few records out of many records meeting your requirement.

- Select first_name,employee_id,salary from employees where department_id=50;
- Select * from employees where last_name='King';
- Select * from employees where hire_date>'17-sep-91';

Comparison operators

- =,>,<,<=,>=,<>

Using comparison operators in where clause

- Select first_name,salary,job_id from employees where salary>10000;
- Select * from employees where salary=24000;

Using other comparison operators

- Between … and …
- In
- Like
- Is null

Using between …and ….

- Use between operator to select rows based on range of values.
- Select first_name,salary from employees where salary between 5000 and 10000;

Using IN operator

- Use IN operator with the condition to check for values in the list.
- Select employee_id,first_name,salary,manager_id from employees where manager_id in(100,101,200);

Using LIKE operator

- User LIKE operator to search a pattern in string values.
- % denotes one or many characters.
- _ denotes one character.
- Select first_name,salary from employees where first_name like ‘S%’;
- Select first_name,salary from employees where first_name like ‘%t_’;

Using IS NULL operator

- Use IS NULL operator to search values
- Select * from employees where manager_id is null;

Logical operators

- AND
- OR
- NOT

Using AND operator

- Select first_name,salary,job_id from employees where salary>5000 AND job_id like '%REP%;

Using OR operator

- Select first_name,salary,job_id from employees where salary>5000 OR job_id like '%REP%;

Using NOT operator

- Select first_name,salary,job_id from employees where job_id not in ('SA_REP','IT_PROG');

Operator precedence

- Arithmetic operators
- Concatenation operator
- Comparison operators
- IS [NOT] NULL, LIKE, [NOT] IN
- [NOT] BETWEEN
- NOT logical condition
- AND logical condition
- OR logical condition

ORDER BY clause

- Sorting the rows using ORDER BY clause
  - ASC for ascending order,default
  - DESC for decending order
- Order by clause comes as last clause in select statement
- Select first_name,salary from employees order by salary;
- Select first_name,salary from employees order by salary desc;

Sorting by multiple columns

- Select last_name, department_id, salary from employees order by department_id, salary desc;