CTIS 259
DATA BASE MANAGEMENT SYSTEMS AND APPLICATIONS

DATABASE : Integrated collection of data

DATABASE MANAGEMENT SYSTEM (DBMS): A software system designed to store, manage, and facilitate access to databases.

Information: Data that have a special meaning (e.g., the fax number of CTIS department in Bilkent University)

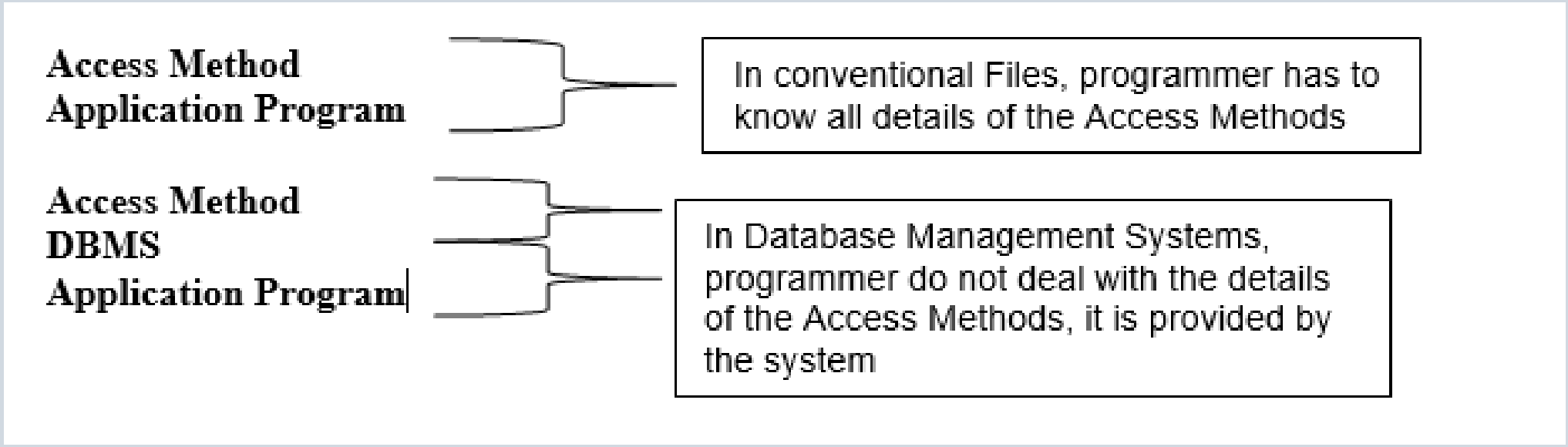
Information System: A collection of software applications and computer hardware that store and manipulate information (e.g., a library information system).

Management Information System (MIS) is a means of supplying information needed by organization.

A **database** is a collection of data, typically describing the activities of one or more organizations.

A **Database Management System (DBMS)** is a collection of programs that enables users to create and maintain database. Data Base Management Systems (DBMS) are the tools for MIS for effectively storing, retrieving and manipulating data. They are a collection of programs, which interface with application programs and manage data on the application programs behalf. DBMS add a layer of software between the application program and the access method.

Database and DBMS software together called as **database system** in general.



Data Model: A description of what kind of information will go into a particular database and how that information will be structured or organized.

Query: A request for some of a database's information content.

The physical database: a collection of files containing the data content

The schema: a specification of the physical database's information content and logical structure

The database engine: software that lets people access and modify the database contents

Records are physical presentations of data a special kind, which are always stored.

An Attribute is a logical unit of data- which by agreement, would have a little meaning if further subdivided. Each attribute carries a single, distinct meaning.

A Field is usually representation of an attribute. It may also relate to some aspect of the view's physical presentation.

DDL (Data Definition Language): A language that provides a set of operations that support the to define the schema of database and access privileges to the database.

DML (Data Manipulation Language): A language that provides a set of operations that support the basic data manipulation operations on the data held in the database.

Database Applications Examples

Enterprise Information:

- Sales: customers, products, purchases
- Accounting: payments, receipts, assets
- Human Resources: Information about employees, salaries, payroll taxes.

Manufacturing: management of production, inventory, orders, supply chain.

Banking and finance:

- customer information, accounts, loans, and banking transactions.
- Credit card transactions
- Finance: sales and purchases of financial instruments (e.g., stocks and bonds; storing real-time market data

Universities: registration, grades

Airlines: reservations, schedules

Telecommunication: records of calls, texts, and data usage, generating monthly bills, maintaining balances on prepaid calling cards

Web-based services

- Online retailers: order tracking, customized recommendations
- Online advertisements

Document databases

Navigation systems: For maintaining the locations of varies places of interest along with the exact routes of roads, train systems, buses, etc.

ADVANTAGES & DISADVANTAGES OF THE DBMS

The following example will be used to explain the problems of conventional file systems and advantages of the DBMS.

In one company, assume that, there are two departments, which are computerized. One of them is Personnel Department and keeps the information about employees as below:

JANE DOE	123-45-6789 (packed decimal)	111 FIRST STREET	P.S. 101	HARWARD	Assistant Director of Personnel
Name	Social Security Number	Address		Education	Job Title

And the other department is Payroll Department and they are keeping information about the Employee as below:

123-45-6789	JANE DOE	111 FIRST STREET	$ 1000 per week	303-193
Social Security Number	Name	Address	Pay rate	Employee number

DISADVANTAGES OF CONVENTIONAL FILE PROCESSING

A-Data Dependence:

Programs, which are using the stored data in conventional file organizations, are **dependent upon data**. If the length of the field of the record changes or the new field is added or one of the existing one deleted(structural changes of the data), the programs accessing to that file should be updated. This is very costly for the big systems. (Since many programs can access to the same file) We called this situation as the "**Data Dependence of the Programs**". **Programs are dependent on the structure of the data.**

ADVANTAGES OF DATA BASE MANAGEMENT SYSTEMS

A- Data Independence:

The database holds not only organization's operational data but in addition it holds a description of this data. For that reason database is also defined as a self-describing collection of integrated data. The description of the data is known as the **system catalogue** (or **data dictionary** or **metadata**-the data about data).

The self-describing nature of database **provides program-data independence**. A database systems contains complete definition or description of the database structure and constraints. This definition is stored in the system **catalog**. The information stored in the catalog is called **meta-data**, and it describes the structure of the primary database. The database which contains this data is called as **data dictionary** .

The DBMS can be tailored to continue to present data in the same format to existing programs and also present data in a new format to new programs being installed. This facility is called data independence. Only the programs that have a need for the additional data need to be changed.

We can change the internal definition of an object without affecting the users of the object, provided the external definition remains same, is known as **data abstraction**.

DISADVANTAGES OF CONVENTIONAL FILE PROCESSING

B- Data Redundancy:

In the given example, address is the redundant data. Redundant data uses extra more space in the storage area and also may cause lack of integrity.

C- Data Consistency:

In the given example, Social Security Number stored in two different file.

----------> Payroll File----- zoned decimal -9 position,

----------->Personnel File --packed decimal -5 position.

Manipulation of the fields for comparison, update, sequence etc. will cause some problems. And data may be **INCONSISTENT**. ***Data inconsistency exists when different and conflicting versions of the same data appear in different places.***

ADVANTAGES OF DATA BASE MANAGEMENT SYSTEMS

B- Controlling Redundancy:

DBMS can store a data element one time and allow multiple programs to access it. Redundancy should be eliminated and only controlled redundancy can be used to represent relationships.

==========> ***No unplanned duplicate data items!!***

C- Consistent Data:

If fields containing data in two different formats are to be compared, the DBMS makes the appropriate conversions before manipulating the data.

DISADVANTAGES OF CONVENTIONAL FILE PROCESSING

D- Data Sequence & Difficulty in Accessing Data:

If we have request such that: "Provide a list of all directors who earn more than $50.000 a year." How can we answer it?

Salary	is in the Payroll File and
Job title	is in the Personnel File

The files must be constantly stored to place data in another sequence to satisfy each reporting requirements. With conventional files, it is not unusual for required data to be in multiple files and to be in a different sequence in each file.

Required information may not be obtained directly by the end-users.

E- Lack of Data Integrity

In the given example, address is the redundant data. In order to obtain "data integrity", when the data is stored twice, it must be updated in both locations whenever it changes. ***Data that display data inconsistency are also referred to as data that lack data integrity.***

ADVANTAGES OF DATA BASE MANAGEMENT SYSTEMS

D- Representing Complex Relationships among Data:

A database may include numerous varieties of data that are interrelated in many ways.

E- Data Integrity:

Data represents what it purposed to. A field provides the correct data to the viewer at all times.

DISADVANTAGES OF CONVENTIONAL FILE PROCESSING

F- Data Security:

Critical data might be obtained and given to competitors, or an incorrect field might be updated accidentally by the unauthorized person. No one can have access to more data than is necessary to do his or her job.

G- Data Isolation:

Since data is scattered in various files, and files may be in different formats, it is difficult to write new application programs to retrieve the appropriate data. Also no one can isolate part of the files records or fields from the unauthorized person.

H- Multiple users (Not provided):

Interaction of concurrent updates may result in inconsistent data (Lost update problems etc.). Also in many conventional file systems, it is not possible to use the same file by more then one people at the same time.

ADVANTAGES OF DATA BASE MANAGEMENT SYSTEMS

F- Restricted Unauthorized Access (Data Security):

DBMS examines each request from on application program to ensure that the program has the authority to manipulate the data it has requested.

===> ***Data is secure from unauthorized access!!***

G- Data Isolation:

Data is isolated as part of the files, records or fields when it is updated by any program from the other programs which will update them until the first program completes.

H- Sharable:

Data files are sharable and concurrency control mechanisms provides concurrent update environment by protecting consistency of data.

DISADVANTAGES OF CONVENTIONAL FILE PROCESSING

I- Spend too Much Space for Redundant Data:

Sometimes there is no way to avoid redundant data.

J- Only Operating System Backup and Recovery Features Available:

These features not adequate enough for very important data.

ADVANTAGES OF DATA BASE MANAGEMENT SYSTEMS

I- Economy of Scale:

Since the applications are concentrated in one location, larger & more powerful computers and secondary storage devices can be purchased. Another benefit is concentration of technical expertise.

J- Providing Backup and Recovery

K- Providing Persistence Storage for Program Objects and Data Structures

L- Providing Multiple User Interfaces:

Form-style interfaces and menu-driven interfaces (known as graphical User Interfaces (GUI) and WWW access are varying types of user interfaces.

DISADVANTAGES OF DATA BASE MANAGEMENT SYSTEMS

A- Size: Programming is very complex and requires many instructions. Some DBMS's require over 2 million bytes of storage.

B- Complexity : The personnel must understand the inner workings of the system to degree that will provide the best service to the business enterprise. Adequate training required choosing optimal way of storing, retrieving, manipulating and protecting the data.

C- Cost : The size of DBMS contributes to its relatively high cost.

D- Hardware Requirements: To execute the instructions to provide the service of the DBMS, the instructions must reside in real storage. Virtual storage may be used. and also the horsepower of the current computer should be increased.

E- Higher Impact of a Failure: The systems are concentrated into one host machine. So the link problem or failure in host cause all the sides to be down.

DBMS FUNCTIONS

Data Dictionary Management: The DBMS requires that definitions of the data elements and their relationships (metadata) be stored in a data dictionary.

Data Storage Management: A modern DBMS systems provides storage not only for the data but also for related data entry forms or screen definitions, report definitions, data validation rules, procedural code, structures to handle video and picture formats and so on.

Data Transformation and Presentation: Buy maintaining data independence, the DBMS translates logical requests into commands that physically locate and retrieve the requested data.

Security Management: The DBMS creates a security system that enforces user security and data privacy within the database.

Multiuser Access Control: The DBMS uses sophisticated algorithms to ensure that multiple users can access the database concurrently without compromising the integrity of the database. (Transaction management and Concurrency control)

Backup and Recovery Management

Data Integrity Management The DBMS promotes and enforces integrity rules o eliminate problems, thus minimizing data redundancy and maximizing data consistency.

Database access languages and application programming interfaces (DDL/DML)

Database Communication Interfaces: The DBMS provide communications functions to access the database through the internet or within the computer network environment.

Database Engine

A database system is partitioned into modules that deal with each of the responsibilities of the overall system. The functional components of a database system can be divided into

- The storage manager,
- The query processor component,
- The transaction management component.

Storage Manager

A program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.

The storage manager is responsible to the following tasks:

- Interaction with the OS file manager
- Efficient storing, retrieving and updating of data\

The storage manager components include:

- Authorization and integrity manager
- Transaction manager
- File manager
- Buffer manager

The storage manager implements several data structures as part of the physical system implementation:

- Data files --store the database itself
- Data dictionary --stores metadata about the structure of the database, in particular the schema of the database.
- Indices --can provide fast access to data items. A database index provides pointers to those data items that hold a particular value.

Query Processor

The query processor components include:

- DDL interpreter --interprets DDL statements and records the definitions in the data dictionary.
- DML compiler --translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.
  - The DML compiler performs query optimization; that is, it picks the lowest cost evaluation plan from among the various alternatives.
- Query evaluation engine --executes low-level instructions generated by the DML compiler.

Query Processing

1.Parsing and translation
2.Optimization
3.Evaluation

Transaction Management

A **transaction** is a collection of operations that performs a single logical function in a database application. They are online-real time and operates on databases.

Transaction-management component ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures.

Concurrency-control manager controls the interaction among the concurrent transactions, to ensure the consistency of the database.

Database Architecture

Centralized databases

- One to a few cores, shared memory

Client-server

- One server machine executes work on behalf of multiple client machines.

Parallel databases

- Many core shared memory
- Shared disk
- Shared nothing

Distributed databases

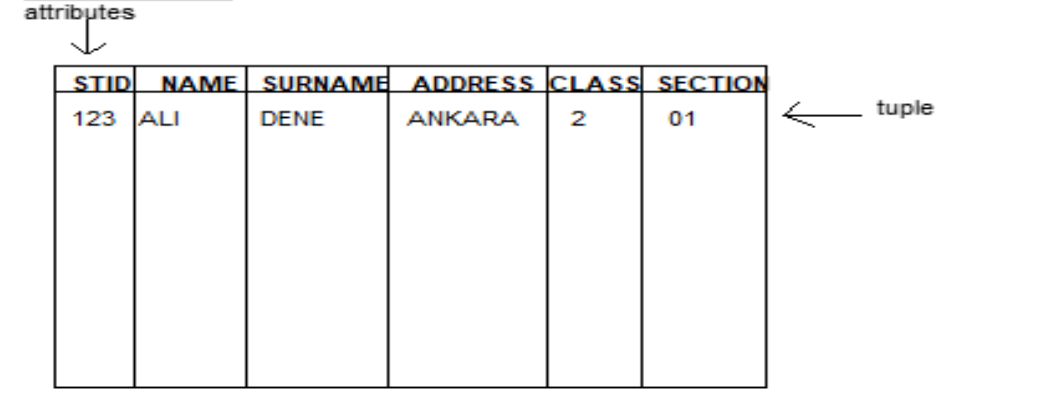
- Geographical distribution
- Schema/data heterogeneity

Data Model and SQL DDL

RELATIONAL DB MODEL

Relational data model represents the database as a collection of tables. There is direct correspondence between concept of the table and mathematical concept of RELATION.

Structure:



Relational_DB Related Term	Programming Lang.(COBOL) Term
Attribute (Column)	Field
Tuple (Row)	Record
Relation (Table)	File

. Each table has a unique name.

. **Domain:** A set of possible values for each attributes called as domain of that attribute.

. **Attributes** are the field of (columns) of the tables. Attribute values are normally required to be atomic; that is indivisible.

. **Database schema:** Logical structure (design) of the database

. **Database Instance:** Data in db at a given instant in time. (a snapshot of the data in the database at a given instant in time.)

. **A tuple** is the collection of values that compose one row of a relation.

KEYS

Every relation has at least one **candidate key**, which is named as **primary key**. And the remaining candidate keys are called as **alternate key**.

KEY TYPE	DEFINITION
Candidate key	Are the attribute(s) that uniquely identify all of the attributes in that relation?
Primary key	A candidate key selected to uniquely identify all attribute values in any given row. Cannot contain null entries. Physical address of a record will depend on that key.
Secondary key	An attribute (or combination of attributes) used strictly for data retrieval purposes. (UNIQUE constraint)
Foreign key	An attribute (or combination of attributes) in one table whose values must either match the primary key in another table or be null.

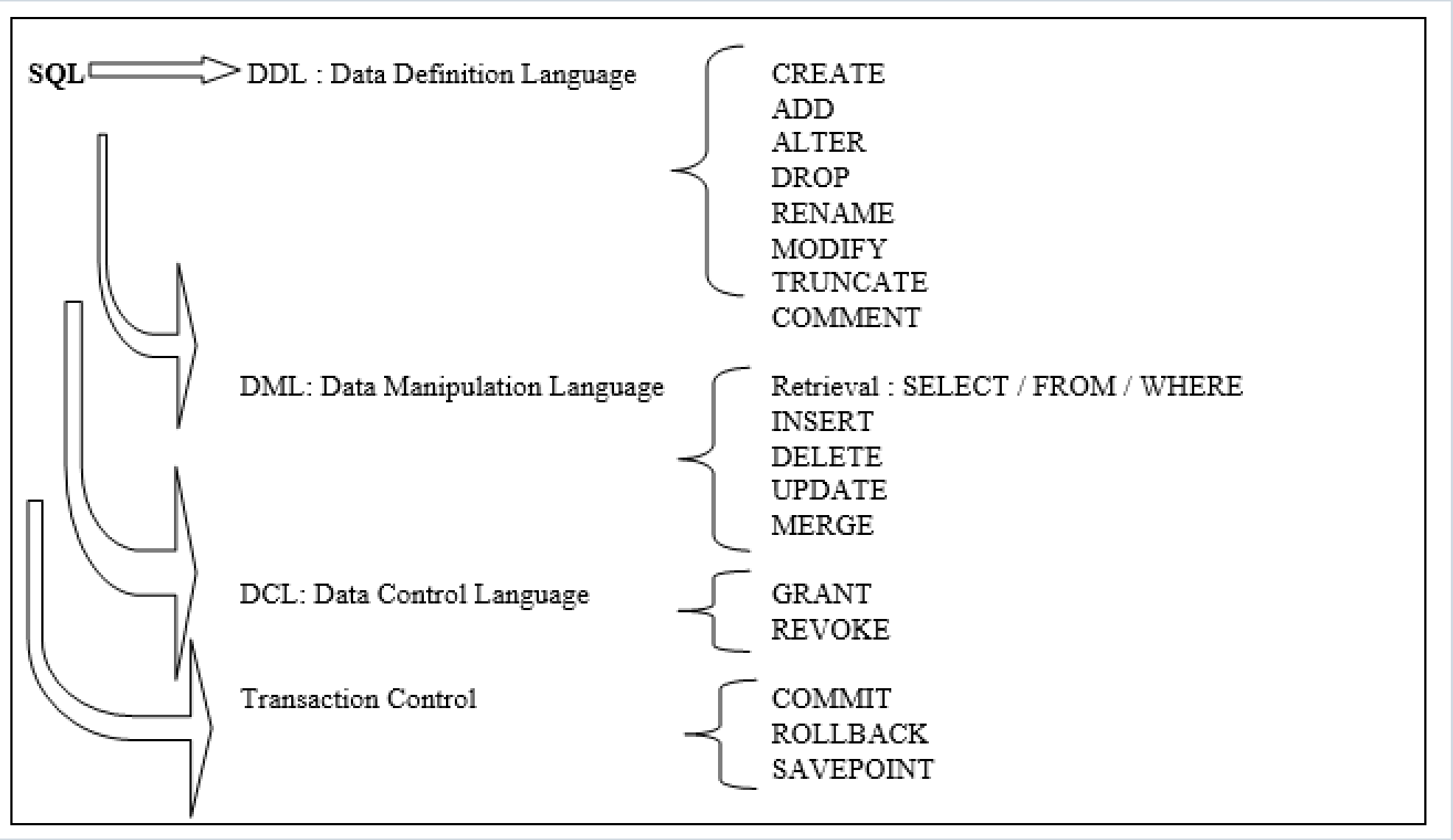
Database Design Steps:

- Requirements Specifications
- Conceptual Design – ER Model
- Logical Design – ER to relational model
- Physical Design
- Normalization

Physical design is deciding on the physical layout of the database. Storage manager is also the part of the physical design part. In the physical design we will not introduce storage management (design and maintenance of the secondary storage), but we will deal only with SQL DDL to create and to manage the tables without defining their storage places etc.

Since we have to use our we will start with SQL DDL & DML, and then we will turn back to Conceptual and Logical Design steps in the second part of the semester.

SQL


SQL
DDL : Data Definition Language
CREATE
ADD
ALTER
DROP
RENAME
MODIFY
TRUNCATE
COMMENT
DML: Data Manipulation Language
Retrieval : SELECT / FROM / WHERE
INSERT
DELETE
UPDATE
MERGE
DCL: Data Control Language
GRANT
REVOKE
Transaction Control
COMMIT
ROLLBACK
SAVEPOINT

Database Objects

Object	Description
Table	Basic unit of storage; composed of rows
View	Logically represents subsets of data from one or more tables
Sequence	Generates numeric values
Index	Improves the performance of some queries
Synonym	Gives alternative name to an object

Naming Rules

Table names and column names must:

- Begin with a letter
- Be 1–30 characters long
- Contain only A–Z, a–z, 0–9, _, $, and #
- Not duplicate the name of another object owned by the same user
- Not be an Oracle server–reserved word

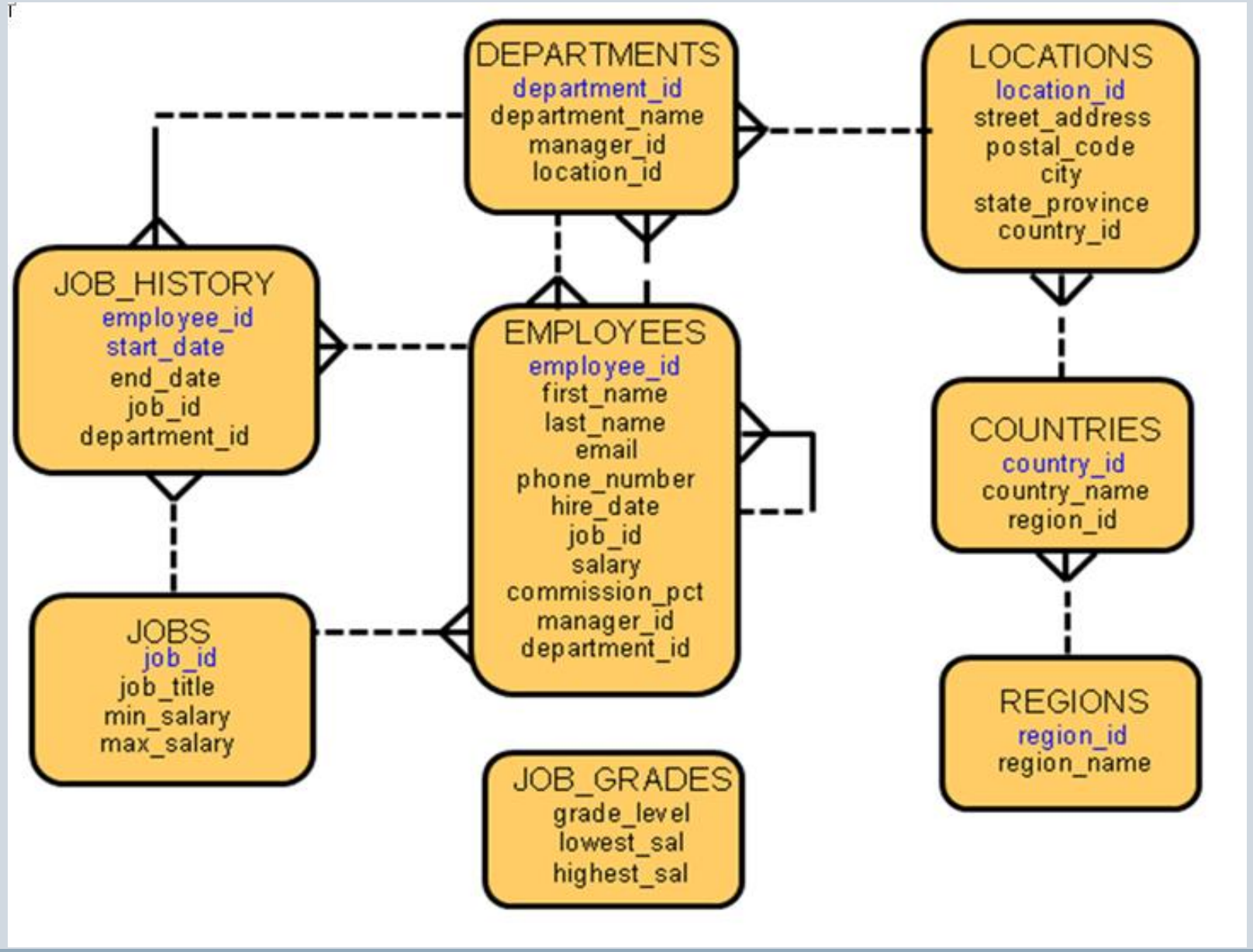
CREATE TABLE: It is used to specify new relation by giving the name and specifying each of its attributes.

DROP TABLE : It is used to delete a relation which is not needed any longer.

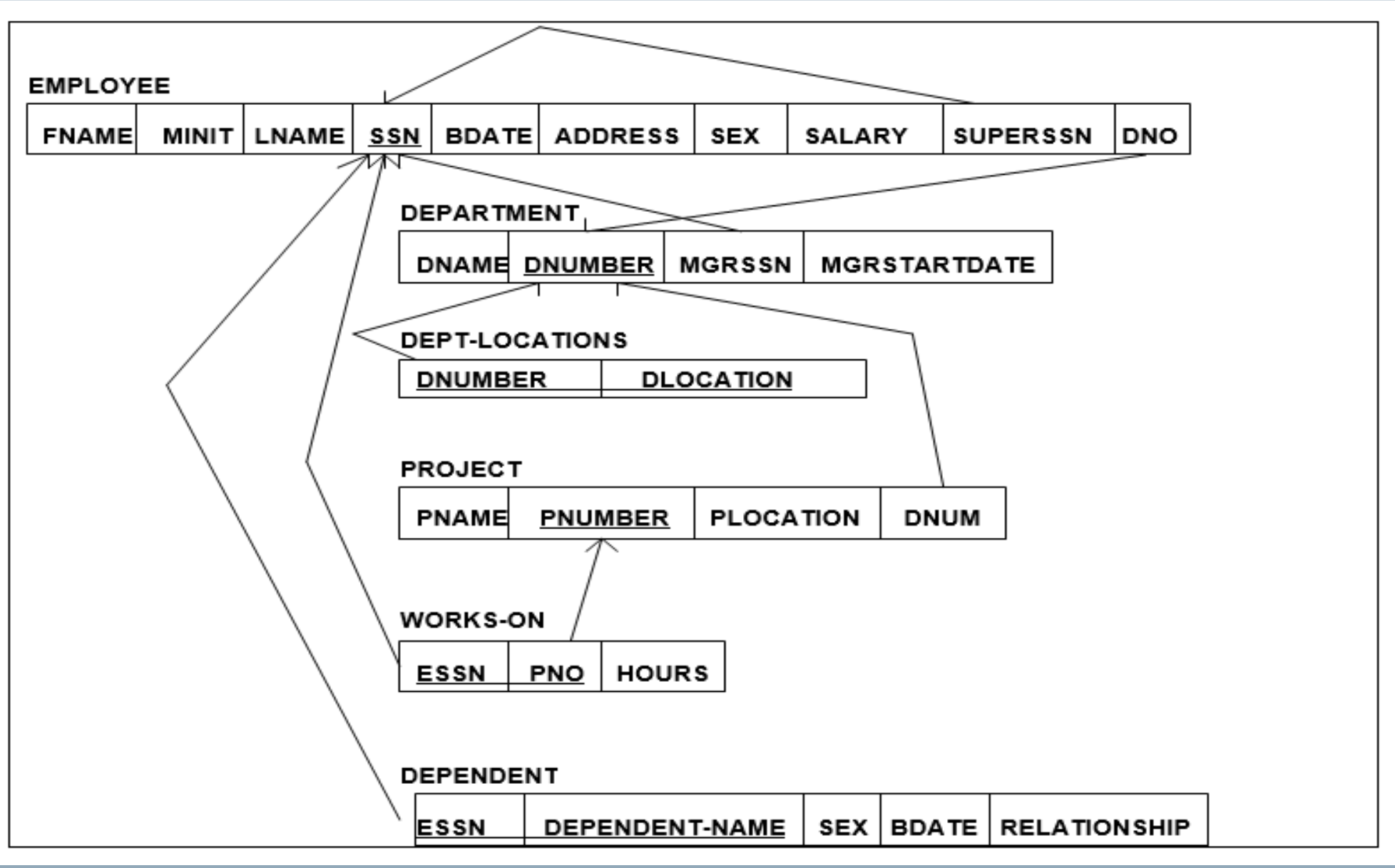
ALTER TABLE: This is used to make changes, additions on table for fields, constraints etc.

DESCRIBE : To see table structure

SAMPLE : HUMAN RESOURCES DATABASE (Ready on your Oracle ORAxx accounts)



SAMPLE: The Company Database Schema:



Data Types

Data Type	Description
VARCHAR2(*size*)	Variable-length character data
CHAR(*size*)	Fixed-length character data
NUMBER(*p*,*s*)	Variable-length numeric data
DATE	Date and time values
LONG	Variable-length character data (up to 2 GB)
CLOB	Character data (up to 4 GB)
RAW and LONG RAW	Raw binary data
BLOB	Binary data (up to 4 GB)
BFILE	Binary data stored in an external file (up to 4 GB)
ROWID	A base-64 number system representing the unique address of a row in its table

Datetime Data Types

You can use several datetime data types:

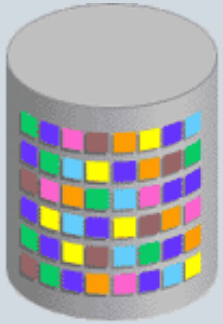
Data Type	Description
TIMESTAMP	Date with fractional seconds
INTERVAL YEAR TO MONTH	Stored as an interval of years and months
INTERVAL DAY TO SECOND	Stored as an interval of days, hours, minutes, and seconds

CREATE TABLE Statement

- You must have:
  - The CREATE TABLE privilege
  - A storage area

```
CREATE TABLE [schema.]table
          (column datatype [DEFAULT expr][, ...]);
```

- You specify:
  - The table name
  - The column name, column data type, and column size



CREATE TABLE: Example

```
CREATE TABLE employees
    ( employee_id    NUMBER(6)
      CONSTRAINT     emp_employee_id   PRIMARY KEY,
      first_name     VARCHAR2(20) ,
      last_name      VARCHAR2(25)
      CONSTRAINT     emp_last_name_nn  NOT NULL,
      email          VARCHAR2(25)
      CONSTRAINT     emp_email_nn      NOT NULL
      CONSTRAINT     emp_email_uk      UNIQUE,
      phone_number   VARCHAR2(20) ,
      hire_date      DATE
      CONSTRAINT     emp_hire_date_nn  NOT NULL,
      job_id         VARCHAR2(10)
      CONSTRAINT     emp_job_nn        NOT NULL,
      salary         NUMBER(8,2)
      CONSTRAINT     emp_salary_ck     CHECK (salary>0) ,
      commission_pct NUMBER(2,2) ,
      manager_id     NUMBER(6)
      CONSTRAINT emp_manager_fk     REFERENCES employees (employee_id),
      department_id  NUMBER(4)
      CONSTRAINT     emp_dept_fk    REFERENCES departments (department_id));
```

DEFAULT Option

- Specify a default value for a column during an insert.

```
... hire_date DATE DEFAULT SYSDATE, ...
```

- Literal values, expressions, or SQL functions are legal values.
- Another column's name or a pseudocolumn are illegal values.
- The default data type must match the column data type.

```
CREATE TABLE hire_dates
        (id          NUMBER(8),
        hire_date DATE DEFAULT SYSDATE);
table HIRE_DATES created.
```

Creating Tables

- Create the table:

```
CREATE TABLE dept
        (deptno      NUMBER(2),
         dname       VARCHAR2(14),
         loc         VARCHAR2(13),
         create_date DATE DEFAULT SYSDATE);
table DEPT created.
```

- Confirm table creation:

```
DESCRIBE dept
```

```
DESCRIBE dept
Name          Null Type
------------ ---- ------------
DEPTNO             NUMBER(2)
DNAME              VARCHAR2(14)
LOC                VARCHAR2(13)
CREATE_DATE        DATE
```

Including Constraints

- Constraints enforce rules at the table level.
- Constraints prevent the deletion of a table and its contents if there are dependencies.
- The following constraint types are valid:
  - `NOT NULL`
  - `UNIQUE`
  - `PRIMARY KEY`
  - `FOREIGN KEY`
  - `CHECK`

Including Constraints

Some other constraints:

- `DEFAULT`
- `BETWEEN .... AND ....`

Constraint Guidelines

- You can name a constraint, or the Oracle server generates a name by using the `SYS_Cn` format.
- Create a constraint at either of the following times:
  - At the same time as the creation of the table
  - After the creation of the table
- Define a constraint at the column or table level.
- View a constraint in the data dictionary.

Defining Constraints

- Syntax:

```
CREATE TABLE [schema.]table
      (column datatype [DEFAULT expr]
      [column_constraint],
      ...
      [table_constraint][,...]);
```

- Column-level constraint syntax:

```
column [CONSTRAINT constraint_name] constraint_type,
```

- Table-level constraint syntax:

```
column,...
  [CONSTRAINT constraint_name] constraint_type
  (column, ...),
```

Defining Constraints

- Example of a column-level constraint:

```
CREATE TABLE employees(
  employee_id  NUMBER(6)
    CONSTRAINT emp_emp_id_pk PRIMARY KEY,
  first_name   VARCHAR2(20),
  ...);
```

1

- Example of a table-level constraint:

```
CREATE TABLE employees(
  employee_id  NUMBER(6),
  first_name   VARCHAR2(20),
  ...
  job_id       VARCHAR2(10) NOT NULL,
  CONSTRAINT emp_emp_id_pk
    PRIMARY KEY (EMPLOYEE_ID));
```

2

NOT NULL Constraint

Ensures that null values are not permitted for the column:

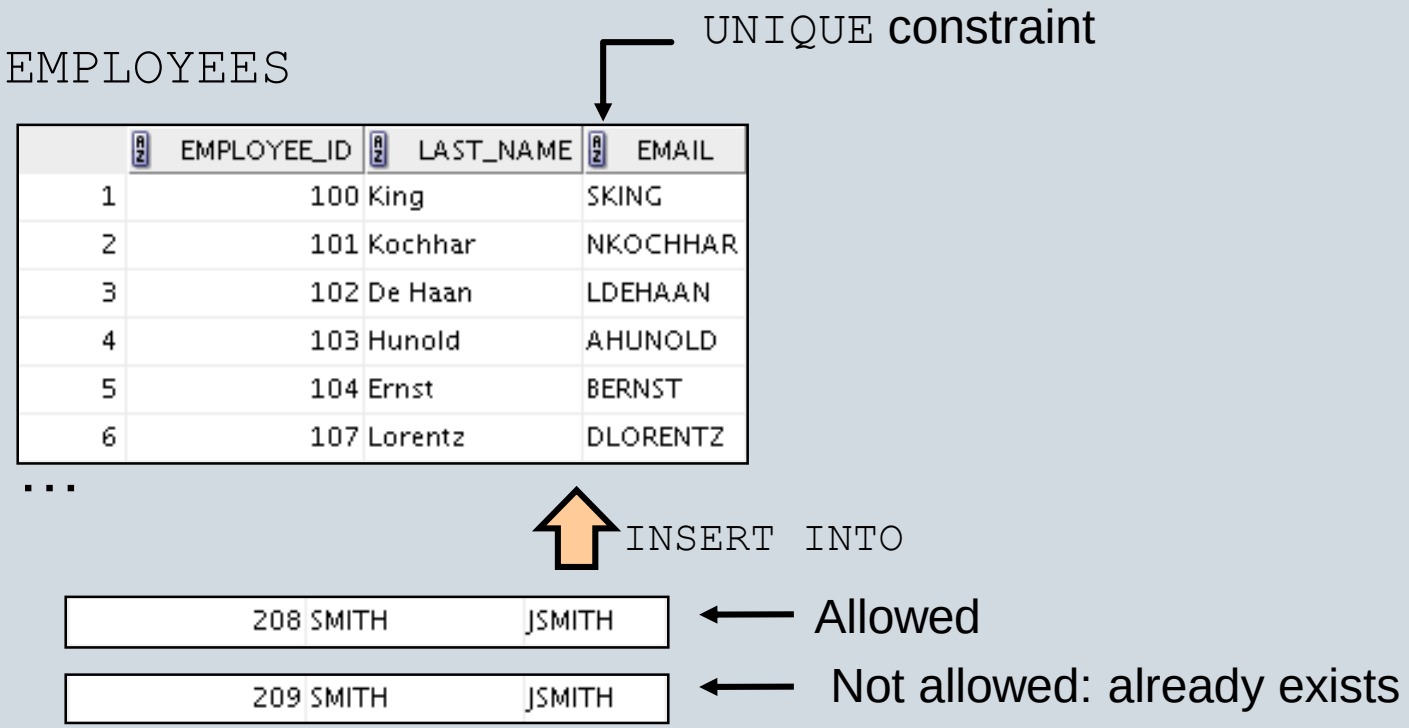
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID	EMAIL	PHONE_NUMBER	HIRE_DATE
100	Steven	King	24000	(null)	90	SKING	515.123.4567	17-JUN-87
101	Neena	Kochhar	17000	(null)	90	NKOCHHAR	515.123.4568	21-SEP-89
102	Lex	De Haan	17000	(null)	90	LDEHAAN	515.123.4569	13-JAN-93
103	Alexander	Hunold	9000	(null)	60	AHUNOLD	590.423.4567	03-JAN-90
104	Bruce	Ernst	6000	(null)	60	BERNST	590.423.4568	21-MAY-91
107	Diana	Lorentz	4200	(null)	60	DLORENTZ	590.423.5567	07-FEB-99
124	Kevin	Mourgos	5800	(null)	50	KMOURGOS	650.123.5234	16-NOV-99
141	Trenna	Rajs	3500	(null)	50	TRAJS	650.121.8009	17-OCT-95
142	Curtis	Davies	3100	(null)	50	CDAVIES	650.121.2994	29-JAN-97
143	Randall	Matos	2600	(null)	50	RMATOS	650.121.2874	15-MAR-98
144	Peter	Vargas	2500	(null)	50	PVARGAS	650.121.2004	09-JUL-98
149	Eleni	Zlotkey	10500	0.2	80	EZLOTKEY	011.44.1344.429018	29-JAN-00
174	Ellen	Abel	11000	0.3	80	EABEL	011.44.1644.429267	11-MAY-96
176	Jonathon	Taylor	8600	0.2	80	JTAYLOR	011.44.1644.429265	24-MAR-98
178	Kimberely	Grant	7000	0.15	(null)	KGRANT	011.44.1644.429263	24-MAY-99
200	Jennifer	Whalen	4400	(null)	10	JWHALEN	515.123.4444	17-SEP-87
201	Michael	Hartstein	13000	(null)	20	MHARTSTE	515.123.5555	17-FEB-96
202	Pat	Fay	6000	(null)	20	PFAY	603.123.6666	17-AUG-97
205	Shelley	Higgins	12000	(null)	110	SHIGGINS	515.123.8080	07-JUN-94
206	William	Gietz	8300	(null)	110	WGIETZ	515.123.8181	07-JUN-94

`NOT NULL` constraint (Primary Key enforces `NOT NULL` constraint.)

`NOT NULL` constraint

Absence of `NOT NULL` constraint (Any row can contain a null value for this column.)

UNIQUE Constraint

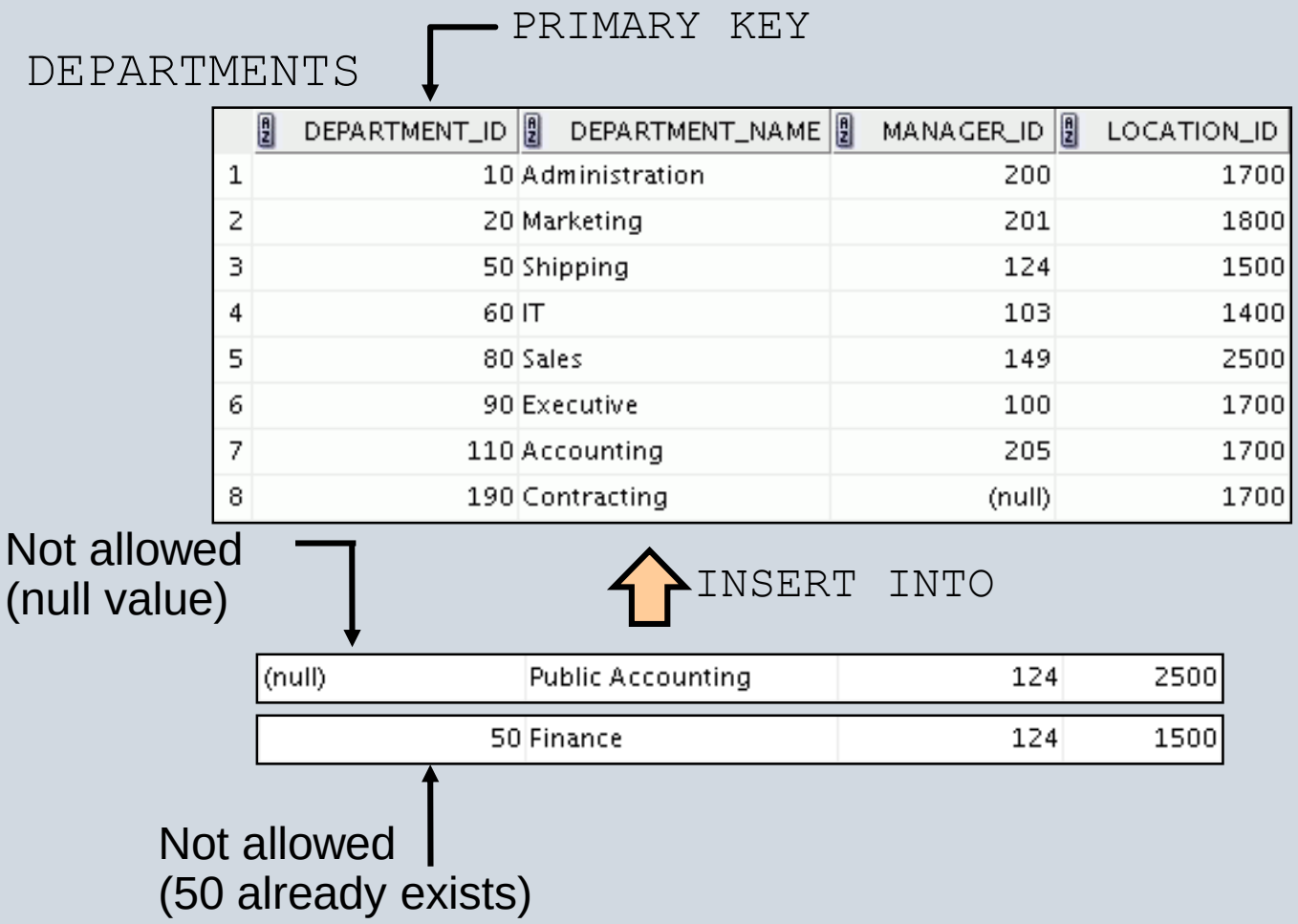


UNIQUE Constraint

Defined at either the table level or the column level:

```
CREATE TABLE employees(
    employee_id      NUMBER(6),
    last_name        VARCHAR2(25) NOT NULL,
    email            VARCHAR2(25),
    salary           NUMBER(8,2),
    commission_pct   NUMBER(2,2),
    hire_date        DATE NOT NULL,
...
    CONSTRAINT emp_email_uk UNIQUE(email));
```

PRIMARY KEY Constraint



`FOREIGN KEY` Constraint: Keywords

- `FOREIGN KEY`: Defines the column in the child table at the table-constraint level
- `REFERENCES`: Identifies the table and column in the parent table
- `ON DELETE CASCADE`: Deletes the dependent rows in the child table when a row in the parent table is deleted
- `ON DELETE SET NULL`: Converts dependent foreign key values to null

CHECK Constraint

- Defines a condition that each row must satisfy
- The following expressions are not allowed:
  - References to CURRVAL, NEXTVAL, LEVEL, and ROWNUM pseudocolumns
  - Calls to SYSDATE, UID, USER, and USERENV functions
  - Queries that refer to other values in other rows

```
..., salary   NUMBER(2)
     CONSTRAINT emp_salary_min
            CHECK (salary > 0),...
```