DATABASE CONNECTIVITY

DATABASE

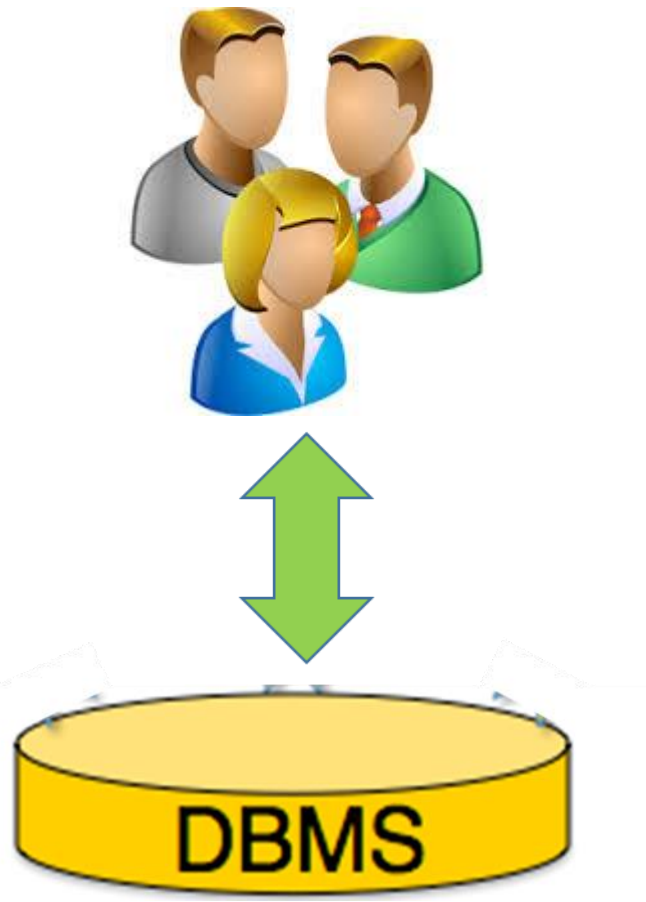
Database is a collection of related data that can be processed to produce information.

A DATABASE MANAGEMENT SYSTEM

It **stores** data in such a way that it becomes easier to **retrieve** and **manipulate**.

1-tier DBMS - Architecture

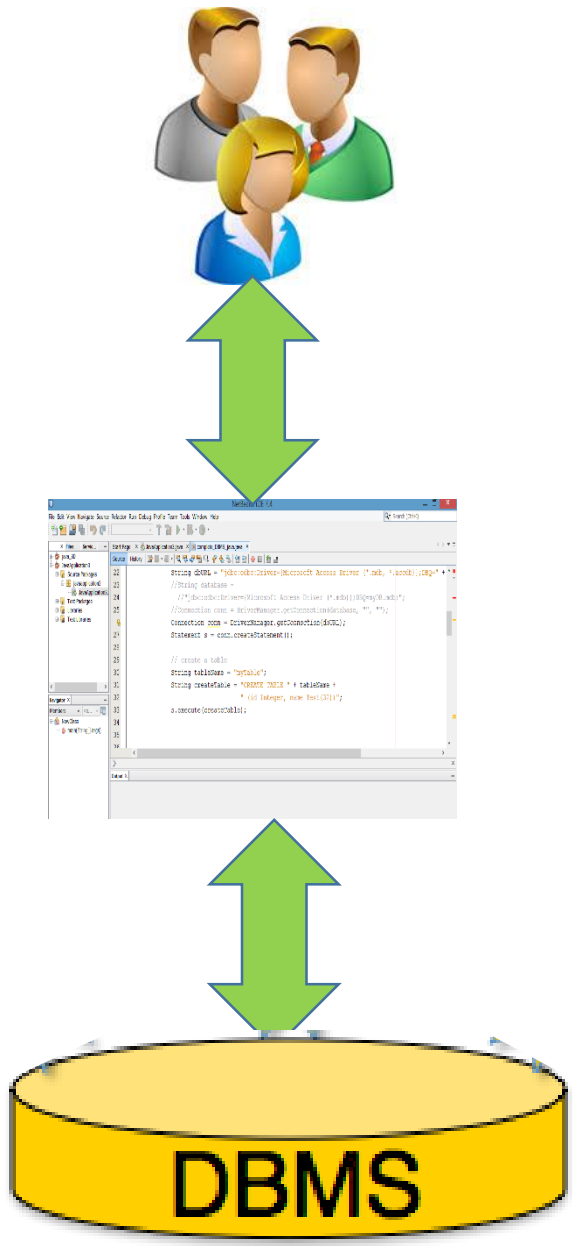
The user directly sits on the DBMS and uses it.
Any changes done here will directly be done on the DBMS itself.



2-tier DBMS - Architecture

The user accesses DBMS through application program.

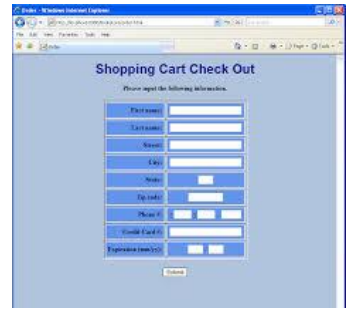
Application program is entirely independent of the database in terms of operation, design, and programming.



3-tier DBMS - Architecture

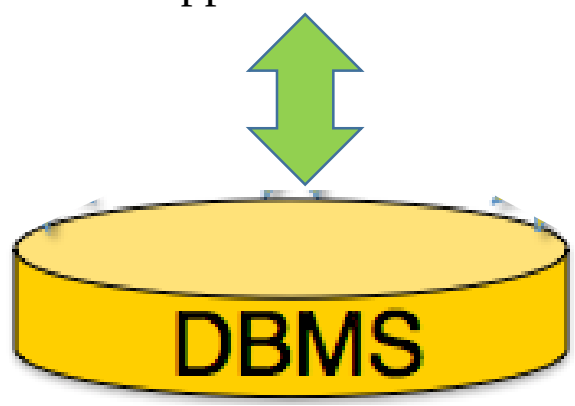
End-users know nothing about any existence of the database beyond this layer.

Acts as a mediator between the end-user and the database.
Application tier presents an abstracted view of the database.



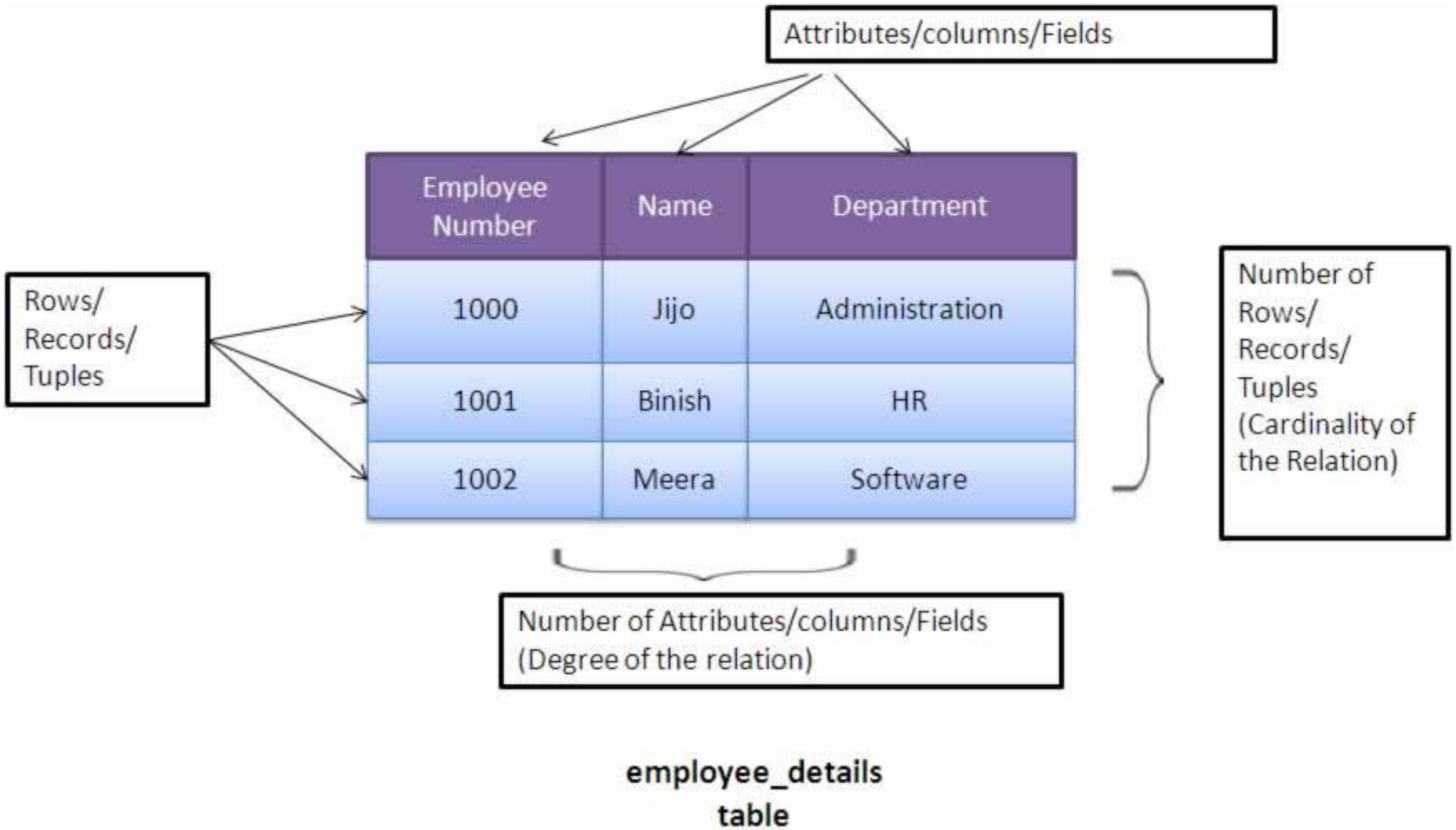
Presentation Tier

Application Tier



Database Tier

RELATIONAL MODEL of DBMS



employee_details table

https://www.google.co.in/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwj7kJSo-MPSAhURTI8KHZ6oBv4QjRwIBw&url=http%3A%2F%2Fwww.careerbless.com%2Fdb%2Frdbms%2Fc1%2Fdatamodels.php&psig=AFQjCNGx5iI4gcTNX-b4CWTZA_rdcB9zzg&ust=1488960324500338

BASIC SQL SYNTAX

CREATE A TABLE

```
CREATE TABLE table_name(
   column1 datatype,
   column2 datatype,
   column3 datatype,
   .....
   columnN datatype,
   PRIMARY KEY( one or more columns )
);
```

```
 CREATE TABLE CUSTOMERS
(
   ID   INT ,
   NAME VARCHAR (20)  ,
   AGE  INT ,
   ADDRESS  CHAR (25) ,
   SALARY   DECIMAL (18, 2),
   PRIMARY KEY (ID)
};
```

ID	NAME	AGE	ADDRESS	SALARY

DROP A TABLE

```
DROP TABLE table_name;
```

```
DROP TABLE CUSTOMERS;
```

ID	NAME	AGE	ADDRESS	SALARY

INSERT VALUES INTO A TABLE

INSERT INTO TABLE_NAME (column1, column2, column3,...columnN)
VALUES (value1, value2, value3,...valueN);

```
INSERT INTO CUSTOMERS
(ID,NAME,AGE,ADDRESS,SALARY)
VALUES (1, 'Ramesh', 32, 'Ahmedabad', 2000.00 );

INSERT INTO CUSTOMERS
(ID,NAME,AGE,ADDRESS,SALARY)
VALUES (2, 'Khilan', 25, 'Delhi', 1500.00 );

INSERT INTO CUSTOMERS
(ID,NAME,AGE,ADDRESS,SALARY)
VALUES (3, 'kaushik', 23, 'Kota', 2000.00 );
```

-
-
-

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	20,000
2	Khilan	25	Delhi	15,000
3	kaushik	23	Kota	56,000
4	Chaitali	25	Mumbai	55,000
5	Hardik	27	Bhopal	25,000
6	Komal	22	MP	10,000
7	Muffy	24	Indore	16,000

DELETE VALUES FROM A TABLE

DELETE FROM table_name WHERE [condition];

DELETE FROM CUSTOMERS WHERE ID = 6;

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	20,000
2	Khilan	25	Delhi	15,000
3	kaushik	23	Kota	56,000
4	Chaitali	25	Mumbai	55,000
5	Hardik	27	Bhopal	25,000
6	Komal	22	MP	10,000
7	Muffy	24	Indore	16,000

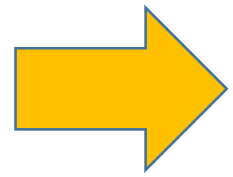
ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	20,000
2	Khilan	25	Delhi	15,000
3	kaushik	23	Kota	56,000
4	Chaitali	25	Mumbai	55,000
5	Hardik	27	Bhopal	25,000
7	Muffy	24	Indore	16,000

UPDATE VALUES OF A TABLE

UPDATE table_name
SET column1 = value1, column2 = value2...., columnN = valueN
WHERE [condition];

```
UPDATE CUSTOMERS
SET ADDRESS = 'Pune'
WHERE ID = 6;
```

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	20,000
2	Khilan	25	Delhi	15,000
3	kaushik	23	Kota	56,000
4	Chaitali	25	Mumbai	55,000
5	Hardik	27	Bhopal	25,000
6	Komal	22	MP	10,000
7	Muffy	24	Indore	16,000



ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	20,000
2	Khilan	25	Delhi	15,000
3	kaushik	23	Kota	56,000
4	Chaitali	25	Mumbai	55,000
5	Hardik	27	Bhopal	25,000
6	Komal	22	Pune	10,000
7	Muffy	24	Indore	16,000

FETCH VALUES FROM A TABLE

SELECT column1, column2, columnN FROM table_name;

SELECT * FROM table_name;

SELECT ID, NAME, SALARY FROM CUSTOMERS;

ID	NAME	SALARY
1	Ramesh	20,000
2	Khilan	15,000
3	kaushik	56,000
4	Chaitali	55,000
5	Hardik	25,000
6	Komal	10,000
7	Muffy	16,000

SELECT * FROM CUSTOMERS;

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	20,000
2	Khilan	25	Delhi	15,000
3	kaushik	23	Kota	56,000
4	Chaitali	25	Mumbai	55,000
5	Hardik	27	Bhopal	25,000
6	Komal	22	Pune	10,000
7	Muffy	24	Indore	16,000

RELATED JAVA CODE

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

This step registers MS access driver by dynamically loading it into memory.

Connection conn = DriverManager.getConnection(dbURL)

Method to create a connection object.

Statement Statement s = conn.createStatement();

Create a statement object from the connection object in order to execute a SQL

s.execute();
Method to **execute** SQL statements

s.close();

conn.close();

END OF CHAPTER