Oracle Day 1

What is RDBMS?

RDBMS stands for relational database management system, which is a program that allows users to create, update, and manage relational databases.

RDBMSs are the most popular database systems for organizations. They are reliable, easy to implement, and capable of storing and retrieving large amounts of data.

Relational databases

Relational databases store data in tables and use Structured Query Language (SQL) to organize and search the data.

How RDBMSs work

RDBMSs use a relational data model to organize data. For example, a small business might use two tables to process orders: one table for customer information, and another for customer orders. The two tables are linked by a common ID column, which allows the database to create a relationship between them.

Some examples of RDBMSs

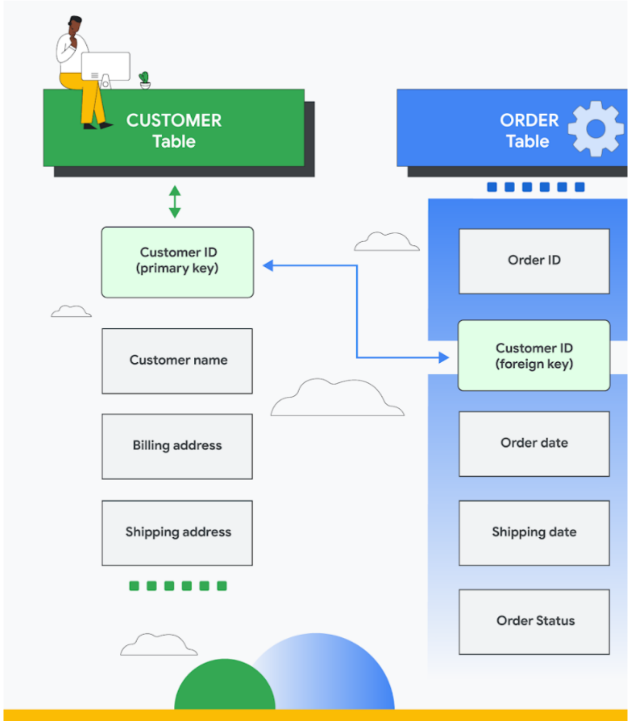
Some well-known RDBMSs include Oracle, MySQL, PostgreSQL, MariaDB, DB2, and Microsoft SQL Server Database.

The relational database model

Developed by **E.F. Codd** from IBM in the **1970**s, the relational database model allows any table to be related to another table using a common attribute. Instead of using hierarchical structures to organize data, Codd proposed a shift to using a data model where data is stored, accessed, and related in tables without reorganizing the tables that contain them.

Think of the relational database as a collection of spreadsheet files that help businesses organize, manage, and relate data. In the relational database model, each “spreadsheet” is a table that stores information, represented as columns (attributes) and rows (records or tuples).

Attributes (columns) specify a data type, and each record (or row) contains the value of that specific data type. All tables in a relational database have an attribute known as the primary key, which is a unique identifier of a row, and each row can be used to create a relationship between different tables using a foreign key—a reference to a primary key of another existing table.



Databases are the cornerstone of any Software Application. You will need one or more databases to develop almost all kinds of software applications: web, enterprise, embedded systems, real-time systems, AI, ML, HPC, blockchain, IoT, and many others.

There are a whopping **343** databases at present. Here I will list popular databases from them

Different databases in the market:

Oracle

MS SQL Server

Teradata

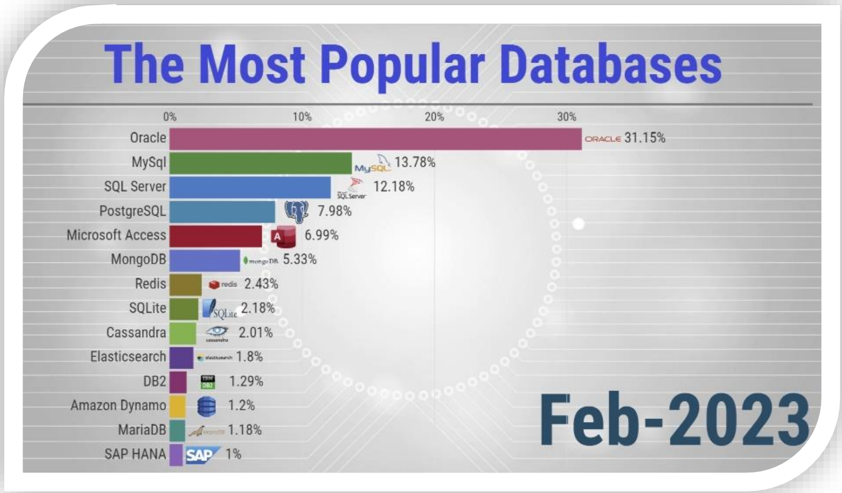
IBM DB2

Sybase

MySQL

PostgreSQL

Netezza



Source: https://statisticsanddata.org

2. Oracle Database

When **Edgar F. Codd** published his revolutionary paper "**A Relational Model of Data for Large Shared Data Banks**" (1970) on the Relational Database Management System (RDBMS), it completely changed the landscape of database Systems. The paper particularly inspired a young Software Engineer **Larry Ellison** (current CTO of Oracle Corporation). He later created the world's first commercially available RDBMS system **Oracle** in 1979. Since then, Oracle remained the leading commercial RDMBS System and dominated the Unix and Linux Systems. Over the last 45 years, Oracle has evolved with time and contributed to the RDBMS and the overall database Systems innovations.

Currently, Oracle is the number one commercially supported database and one of the most widely used RDBMS overall. Its latest release (23ai) has added many innovative features that will make it an attractive option in the coming years.

5 Key Features

- Proprietary RDBMS.
- Offers ACID transactional guarantee. In terms of CAP, it offers immediate Consistency as a single Server.
- Advanced Multi-Model databases supporting Structured Data (SQL), Semi-Structured Data(JSON, XML), Spatial Data, and RDF Store. Offers multiple access pattern depending on the specific Data Model
- Offers Blockchain Tables.
- Supports both OLTP and OLAP workload.

When to Use Oracle

- If a company wants to have a Converged database or Master Database (One database for OLTP and OLAP).
- Traditional transactional workloads with structured (SQL) data, and when ACID transaction guarantee is a key criterion.
- Blockchain Table is required.
- For Data Warehousing.
- A multi-model database including Geospatial Data type is an essential requirement.

When Not to Use Oracle

- If a company wants to save money on a database.
- Multi-Master ACID transaction is a must-have feature.
- Data is Semi-structured, i.e., JSON data with advanced query functions.
- Data is extremely relational (e.g., social media), i.e., Graph-like data.

Oracle As a Service

- Oracle Converged Database
- Amazon RDS for Oracle

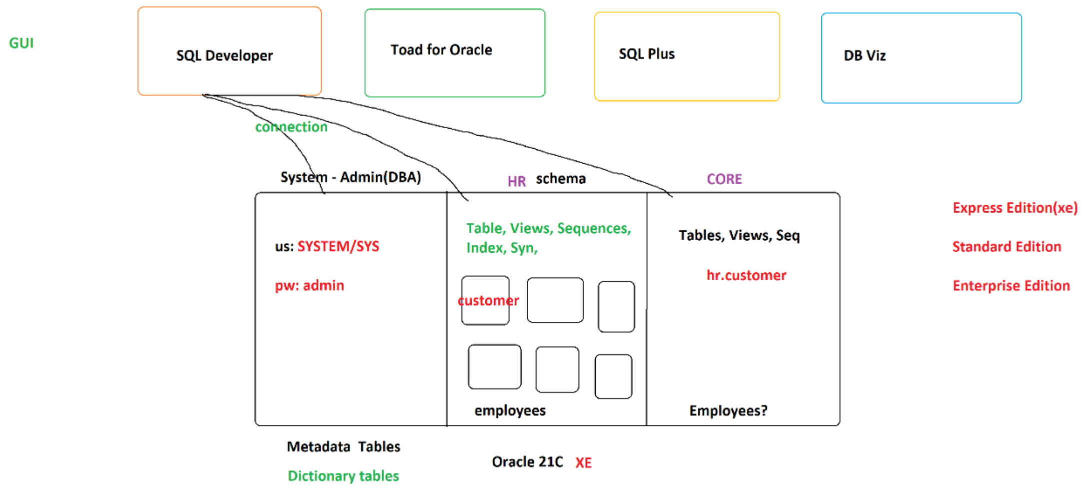
In the past, almost all databases were relational. They used a set data structure, which allowed them to link information from different "tables", using indexes. These data "buckets" could then be linked through a "relationship". **SQL** (Structured Query

Language) is the language used for this kind of database. It provides commands to create, retrieve, update, and delete information stored in the tables.

NoSQL is an acronym for "not only SQL" and refers to a database management system that stores data in a non-relational way. NoSQL databases are different from relational databases, which store data in rule-based tables. NoSQL databases are more flexible and can store a variety of data types, including documents, graphs, key-value, and wide columns

Oracle Set up:

Please watch any one of the below installation videos and install the Oracle database with SQL developer. The software setup up given in the description of the video.



Oracle 11g Installation: https://youtu.be/Jrefk1hOJQg

Oracle 21C Installation: https://youtu.be/xPSbxrMe7VU

Please watch the YouTube video to understand complete SQL concepts and practice below mentioned queries.

Oracle SQL Day 1: https://youtu.be/hLnKjYGr730

SQL -- Structured Query Language

ANSI

Data Types in Oracle

1.Numeric

int

decimal

float

double

number

number(6)	999999
number(8,2)	999999.99
number(2,2)	0.99

2. Character

char	2000
varchar	2000
varchar2	4000

first_name char(10)   ARUN + 6 char (reserved)

first_name varchar2(10)   ARUN + 6 char released

3. Date

date

```
insert   -- format     mm/dd/yyyy
                      dd/mm/yyyy
```

4. LOB

CLOB   -- GB

```
create table test
(
cust_id number(2,2),
cust_name char(10),
cust_name2 varchar(10)
);

drop table test;
```

```
insert into test values(0.89,'Arun','john');

desc test;

select * from test;

select length(cust_name),length(cust_name2) from test;
```

SQL

DDL DML DRL TCL DCL

DDL - Data Definition Language -- Auto Commit

create

alter

rename

truncate

drop

DML - Data Manipulation Language -- User commit

Insert

update

delete

Merge

DRL - Data Retrival language:

select

TCL - Transaction Control Language

commit

rollback

savepoint

DCL - Data Control Language (DBA)

Grant

Revoke

```
create table table_name

(

column_1 data_type,

column_2 data_type,

column_3 data_type,

.

.

.

column_n data_type

);
```

```
create table customer

(

cust_id number(6),

cust_name varchar2(30),

mobile_no  number(10),

dob date,

city varchar2(100),

email_id varchar2(100)

);
```

```
insert into table_name

(column1,column2,column3)
```

```
values
(value1,value2,value3);



create table customer
(
cust_id number(6),
cust_name varchar2(30),
dob date,
mobile number(10),
address varchar2(100)
);


select * from customer;


insert into customer
(CUST_ID,cust_name,dob,mobile,address)
values
(100000,'Arun',to_date('09/12/1992','mm/dd/yyyy'),9090909090,'Chennai');


select * from customer;


rollback;
```

```
commit;


insert into customer
values
(100001,'Kannan',to_date('09/11/2000','mm/dd/yyyy'),8132437493,'Chennai');


insert into customer
values
(100002,'Radha',to_date('09/24/2012','mm/dd/yyyy'),1348374989);
--SQL Error: ORA-00947: not enough values

insert into customer
(CUST_ID,cust_name,dob,mobile)
values
(100002,'Radha',to_date('09/24/2012','mm/dd/yyyy'),1348374989);


commit;


update table_name
set column_name=value
where condition;


update customer
set address='Hydrabad';
```

```
rollback;

select * from customer;

update customer
set address='Hydrabad'
where cust_id=100002;

commit;

-- Add a column

alter table table_name
add column_name data_type;

alter table customer
add zip number(6);

select * from customer;
```

```
--drop a column

alter table table_name
drop column column_name;

alter table customer
drop column address;

-- Rename a table

rename old_table_name to new_table_name;

rename customer to customer_details;

select * from customer_details;

-- rename a column

alter table table_name
rename column old_name to new_name;
```

```
alter table customer_details

rename column mobile to mobile_no;
```

----

1. bkp a table

2. truncate base table

3. modify data type

4. Restore the data

5. drop bkp table
