Database Users

Database administrators – DBA is responsible for authorizing access to the database, for coordinating and monitoring its use, and acquiring software and hardware resources as needed.

Database designers

- identify data to be stored in the database and choosing appropriate structures to represent and store the data. Most of these functions are done before the database is implemented and populated with the data.

- It is the responsibility of the database designers to communicate with all prospective users to understand their requirements and come up with a design that meets these requirements. Database designers interact with all potential users and develop views of the database that meet the data and processing requirements of these groups.

- The final database must support the requirements of all user groups.

End Users

1. Casual End Users – occasionally access, may need different information each time. Use query language to specify requests.

2. Naïve or parametric end users – main job is to query and update the database using standard queries and updates. These canned transactions have been carefully programmed and tested.

3. Sophisticated end users – engineers, scientists, analysts who implement applications to meet their requirements.

4. Stand-alone users – maintain personal databases using ready-made packages.

Primary Roles of Database Administrator

A database administrator's (DBA) primary job is to ensure that data is available, protected from loss and corruption, and easily accessible as needed. Below are some of the chief responsibilities that make up the day-to-day work of a DBA.

DSP deliver an outsourced DBA service, providing Oracle Support and SQL Server Support; whilst mindset and toolset may be different, whether a database resides on premise or in a Public / Private Cloud, the role of the DBA is not that different.

1. Software installation and Maintenance

- A DBA often collaborates on the initial installation and configuration of a new Oracle, SQL Server etc. database.

- The system administrator sets up hardware and deploys the operating system for the database server, then the DBA installs the database software and configures it for use.

- As updates and patches are required, the DBA handles this on-going maintenance. And if a new server is needed, the DBA handles the transfer of data from the existing system to the new platform.

- A **patch** is a set of changes to a computer program or its supporting data designed to update, fix, or improve it. This includes fixing security vulnerabilities and other bugs, with such patches usually being called **bug fixes or bug fixes**.

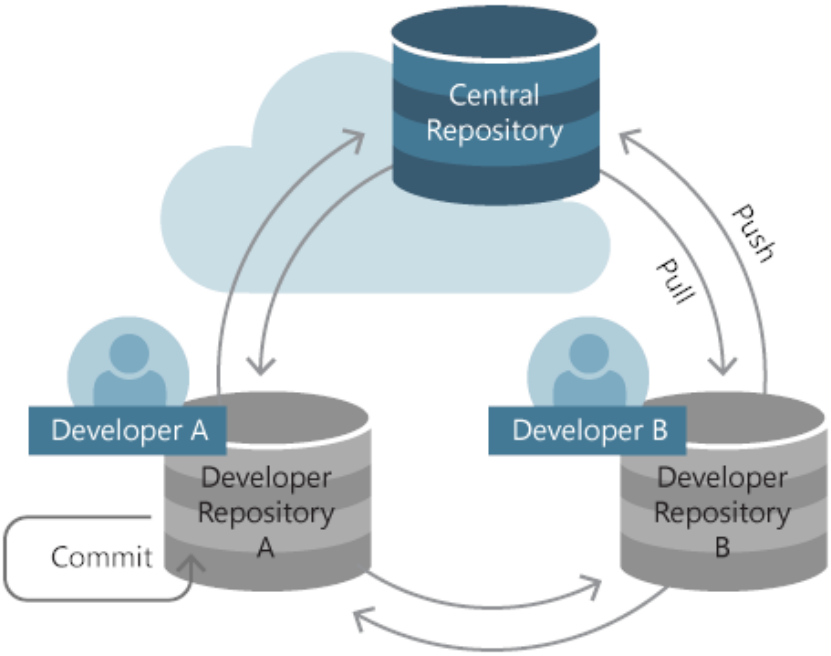
- A major component of a bug tracking system is a database that records facts about known bugs.

2. **Data Extraction, Transformation, and Loading**

- Known as ETL, data extraction, transformation, and loading refers to efficiently importing large volumes of data that have been extracted from multiple systems into a data warehouse environment.

- This external data is cleaned up and transformed to fit the desired format so that it can be imported into a central repository.

- A **database repository** is a logical, but also sometimes physical grouping of data from related but separate databases. This is usually done when there is a 'higher purpose' for the data, but the data items needed to do this reside on **different databases**



3. Specialized Data Handling

- Today's databases can be massive and may contain unstructured data types such as images, documents, or sound

and video files. Managing a very large database (VLDB) may require higher-level skills and additional monitoring and tuning to maintain efficiency.

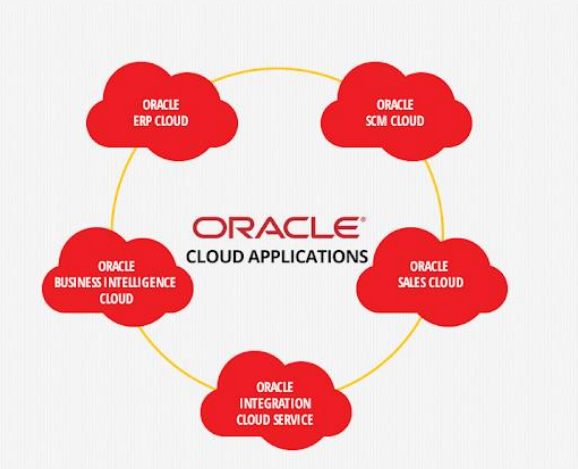
4. Database Backup and Recovery

- DBAs create backup and recovery plans and procedures based on industry best practices, then make sure that the necessary steps are followed. Backups cost time and money, so the DBA may have to persuade management to take necessary precautions to preserve data.

- System admins or other personnel may actually create the backups, but it is the DBA's responsibility to make sure that everything is done on schedule.

- In the case of a server failure or other form of data loss, the DBA will use existing **backups** to restore lost information to the system. Different types of failures may require different recovery strategies, and the DBA must be prepared for any eventuality.

- With technology change, it is becoming ever more typical for a DBA to backup databases to the cloud.



5. Security

- A DBA needs to know potential weaknesses of the database software and the company's overall system and work to minimize risks. No system is one hundred per cent immune to attacks, but implementing best practices can minimize risks.

- In the case of a security breach or irregularity, the DBA can consult audit logs to see who has done what to the data. Audit trails are also important when working with regulated data.

6. Authentication

- Setting up employee access is an important aspect of database security. DBAs control who has access and what type of access they are allowed.

- For instance, a user may have permission to see only certain pieces of information, or they may be denied the ability to make changes to the system.

7. Capacity Planning

- The DBA needs to know how large the database currently is and how fast it is growing in order to make predictions about future needs.

- Storage refers to how much room the database takes up in server and backup space. Capacity refers to usage level. If the company is growing quickly and adding many new users, the DBA will have to create the capacity to handle the extra workload.

8. Performance Monitoring

- Monitoring databases for performance issues is part of the on-going system maintenance a DBA performs. If some part of the system is slowing down processing, the DBA may need to make configuration changes to the software or add additional hardware capacity.

- Many types of monitoring tools are available, and part of the DBA's job is to understand what they need to track to improve the system.

- 3rd party organizations can be ideal for outsourcing this aspect, but make sure they offer modern DBA support.

9. Database Tuning

- Performance monitoring shows where the database should be tweaked to operate as efficiently as possible. The physical configuration, the way the database is indexed, and how queries are handled can all have a dramatic effect on database performance. With effective monitoring, it is possible to proactively tune a system based on application and usage instead of waiting until a problem develops.

10. Troubleshooting

- DBAs are on call for troubleshooting in case of any problems. Whether they need to quickly restore lost data or correct an issue to minimize damage, a DBA needs to quickly understand and respond to problems when they occur.

Database Design and ER Diagram

The database design process can be divided into six steps. The ER model is most relevant to the first three steps:

(i) **Requirements Analysis**: The very first step in designing a database application is to understand what data is to be stored in the database, what applications must be built on top of it, and what operations are most frequent and subject to performance requirements. In other words, we must find out what the users want from the database.

(ii) **Conceptual Database Design**: The information gathered in the requirements analysis step is used to develop a high-level description of the data to be stored in the database, along with the constraints that are known to hold over this data. This step is often carried out using the ER model, or a similar high-level data model, and is discussed in the rest of this module.

(iii**) Logical Database Design**: We must choose a DBMS to implement our database design, and convert the conceptual database design into a database schema in the data model of the chosen DBMS. We will only consider relational DBMS's, and therefore, the task in the logical design step is to convert an ER schema into a relational database schema.

Beyond the ER Model

ER modeling is sometimes regarded as a complete approach to designing a logical database schema. This is incorrect because the **ER diagram is just an approximate description of the data, constructed through a very subjective evaluation of the information collected during requirements analysis.** The remaining three steps of database design are briefly described below:

(iv) Schema Refinement: The fourth step in database design is to analyze the collection of relations in our relational database schema to identify potential problems, and to refine it.

(v) Physical Database Design: In this step we must consider typical expected workloads that our database must support and further refine the database design to ensure that it meets desired performance criteria.

(vi) Security Design: In this step, we identify different user groups and different roles played by various users (e.g., the development team for a product, the customer support representatives, and the product manager).

Entities, Attributes & Entity Sets

An entity is an object that exists and is distinguishable from other objects.

– Example: *student, department, employee and branch*

Entities have attributes, which defines the property of an entity

– Example: *student has names and rollno.*

There are different types of attributes which are categorized as follows:

i) **Simple Attributes:** having atomic or indivisible values. For e.g.

Dept–a string, *PhoneNumber*–an eight-digit number.

ii) **Composite Attributes**: having several components in the value. For

e.g.: *Qualification* with components *(DegreeName, Year, UniversityName).*

iii) ***Derived Attributes****: Attribute value is dependent on some other*

attribute. For e.g.:*Age* depends on *DateOfBirth*. So age is a derived attribute.

iv) **Single-valued Attributes**: having only one value rather than a set

of values. For e.g. *PlaceOfBirth*–single string value.

v) **Multi-valued Attributes**: having a set of values rather than a single

value. For e.g., *CoursesEnrolled* attribute for student, *EmailAddress*

attribute for student, *PreviousDegree* attribute for student.