Database Applications

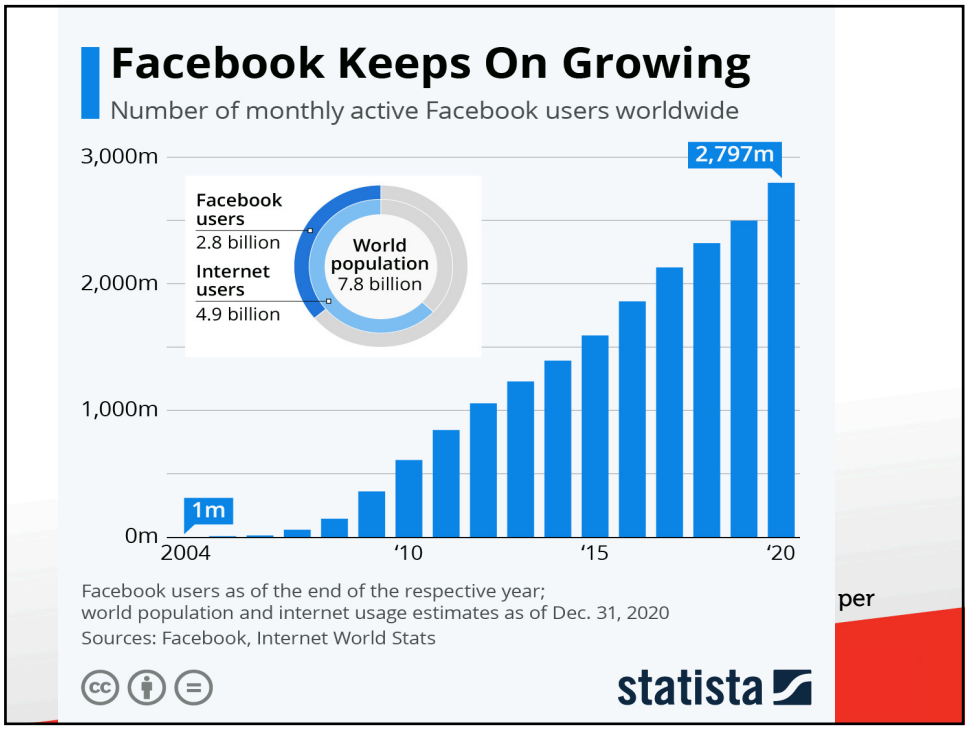
Lecture 1: Introduction and Database Architecture

Santha Sumanasekara

July 2022

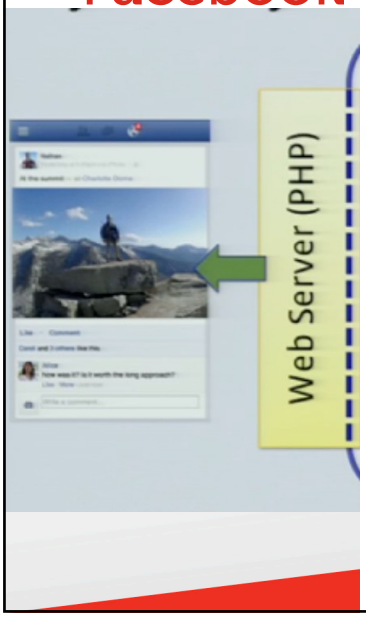
RMIT UNIVERSITY

1

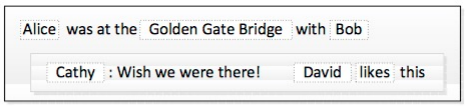


2

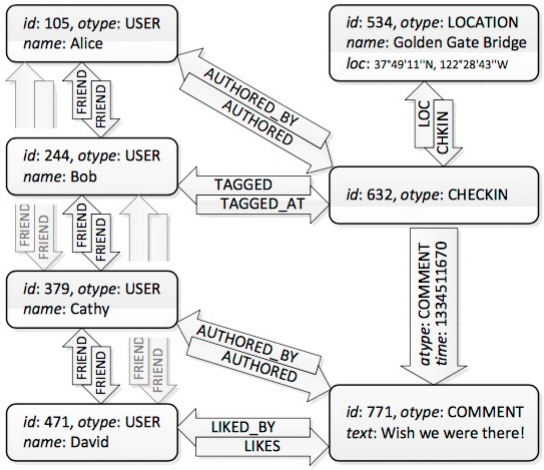
Facebook



a)



b)



3

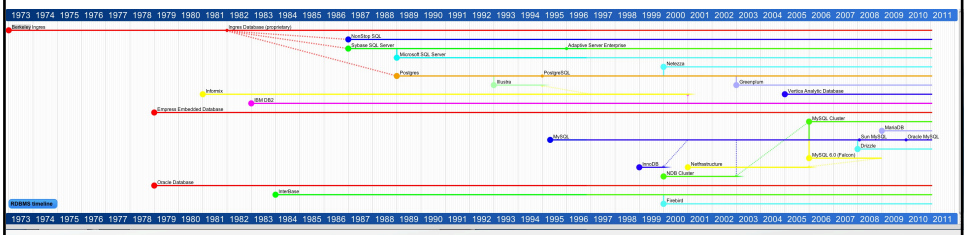
Data Everywhere

- Storing large volumes of data in a methodical way is essential
- Traditional storage methods – flat files, spreadsheets – became insufficient and limited in many ways
- To meet the ever-increasing demand, since 1970s, databases and database management systems have been in use.

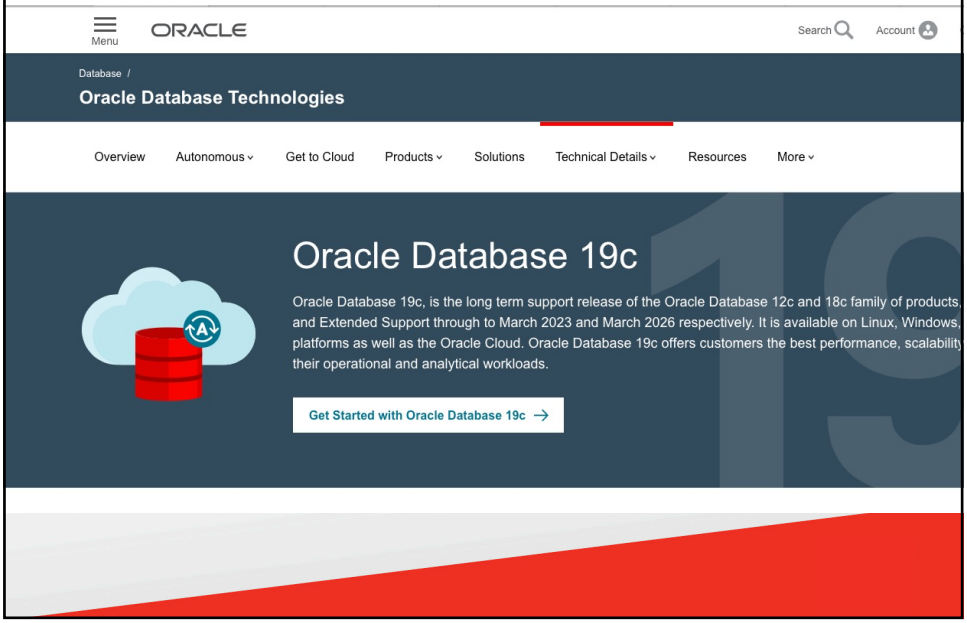
4

Which database products?

- Evolution timeline at
- http://upload.wikimedia.org/wikipedia/commons/2/22/RDBMS_timeline.svg

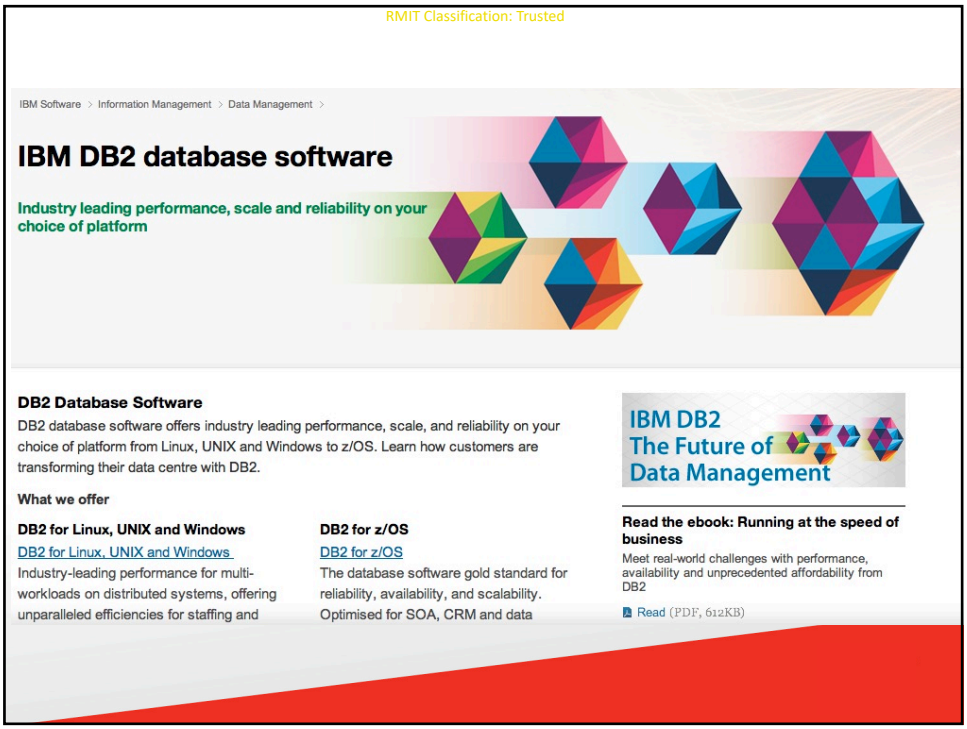


5

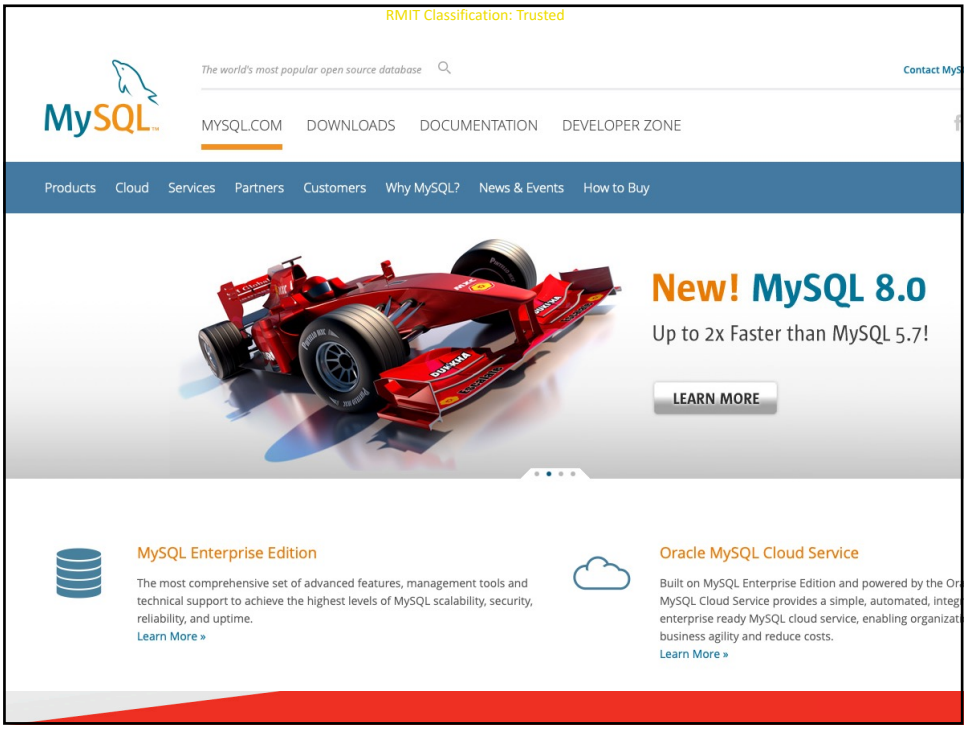


6

7



8



9

A new approach – noSQL

- A NoSQL (originally referring to "non SQL" or "not-only relational" database provides a mechanism for storage and retrieval of data which is modelled in means other than tables.
- a surge of popularity in the early twenty-first century, triggered by the needs of Web 2.0 companies such as Facebook, Google and Amazon.com.
- https://en.wikipedia.org/wiki/NoSQL

10

A new approach – noSQL

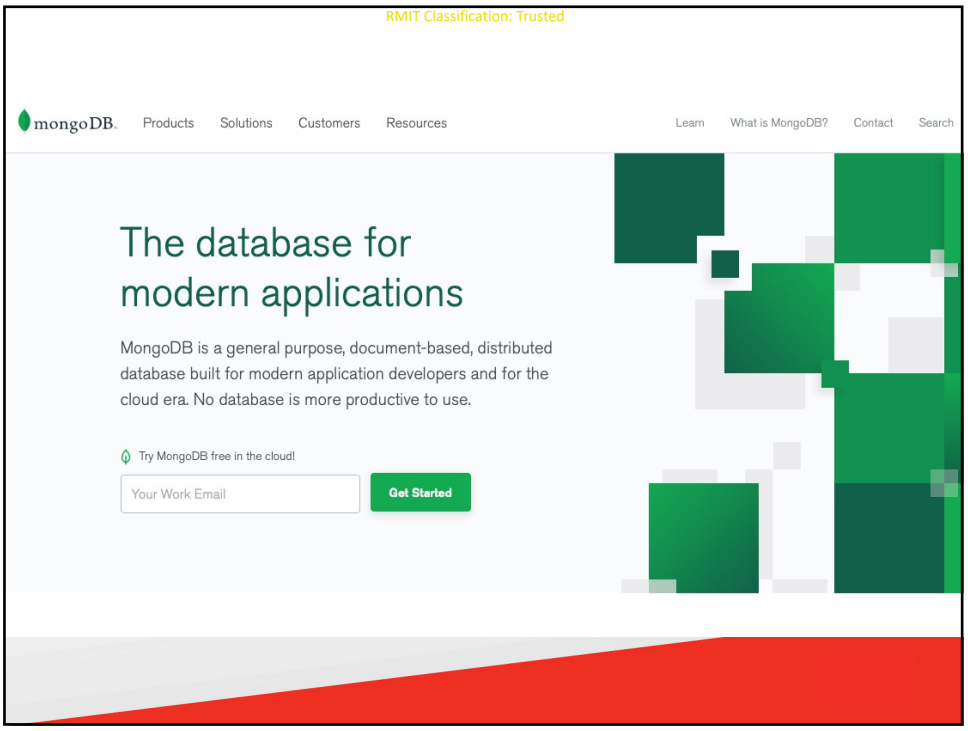
- Source: https://www.improgrammer.net/top-10-databases-shoul learn-2015/

11

A new approach – noSQL

- More informative video on noSQL
  - https://www.youtube.com/watch?v=rRoy6I4gKWU NoSQL vs SQL
- Intro to MongoDB Tutorial:
  - https://www.calebcurry.com/intro-to-mongodb/

12



13

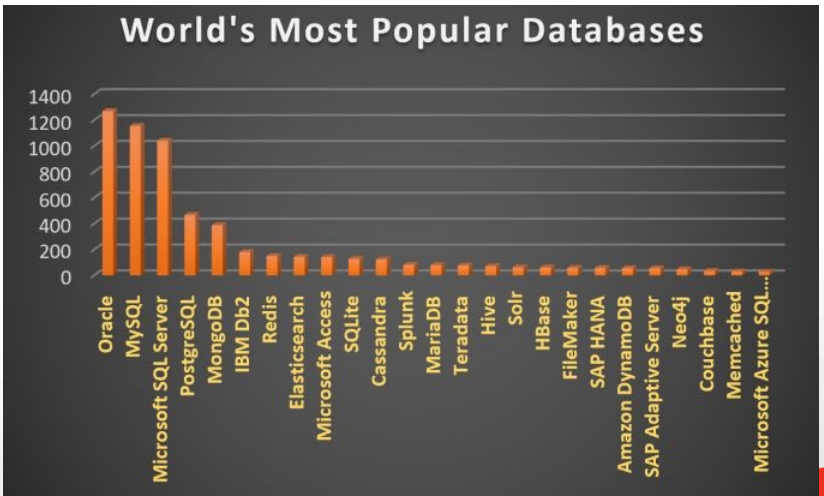
What database have you used?

14

Which database is most commonly used today?

15

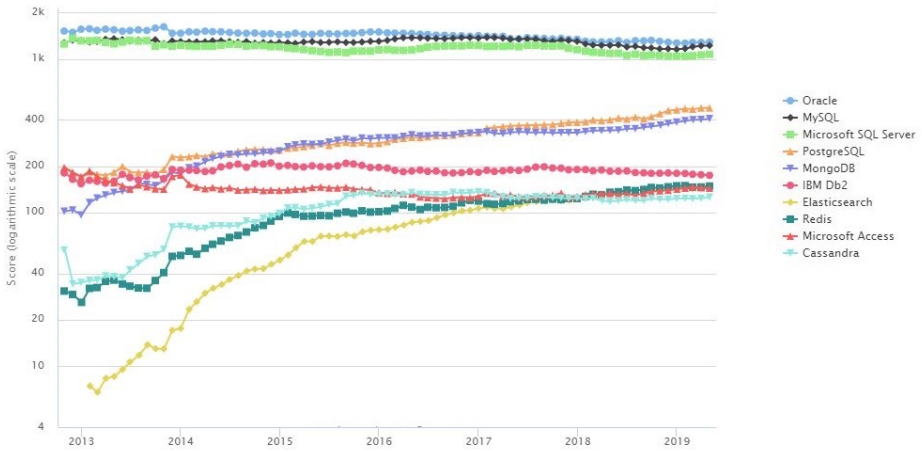
Which database is most commonly used?



- Source: https://www.c-sharpcorner.com/article/what-is-the-most-popular-database-in-the-world/

16

Which database is most commonly used?



- Source: https://www.improgrammer.net/top-10-databases-should-learn-2015/

17

Which database products?

- https://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems

18

What is a Database Application?

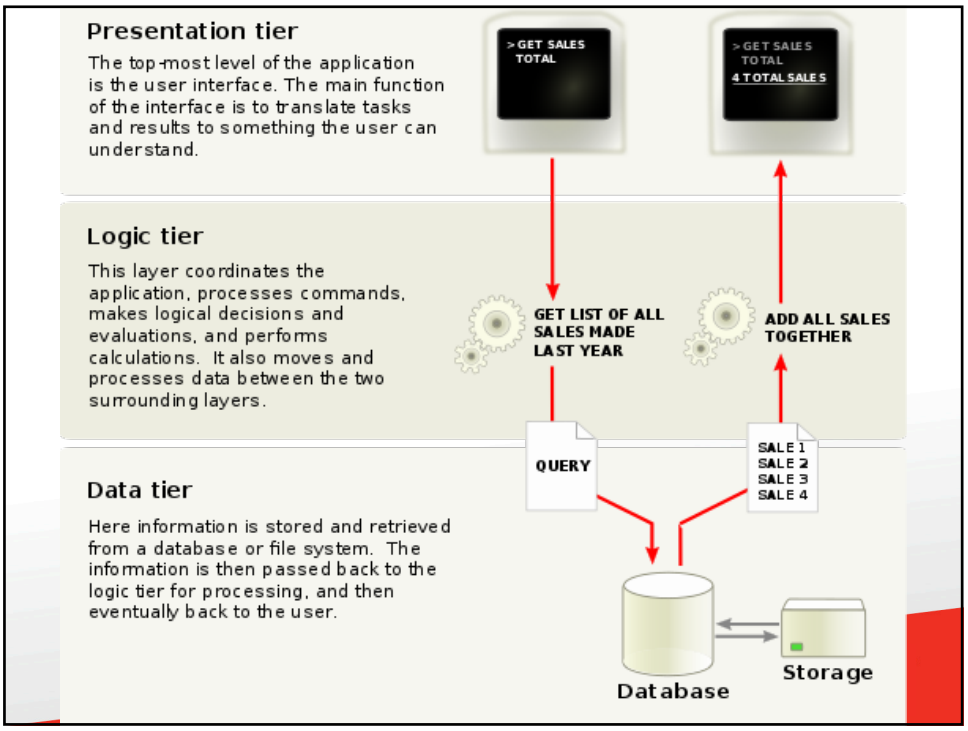
- A database application is a computer program whose primary purpose is entering and retrieving information from a database.
- Early database applications were desktop applications, such as airline reservation systems (e.g. SABRE)
- Starting in the mid-1990s it became more common to build database applications with a Web interface.
- Examples of early database applications with Web interfaces include amazon.com, which used the Oracle as the back-end database.
- Modern database applications like Facebook still uses MySQL and PHP, however, in order to meet the demands of handling over two billion active users, it uses other layers such as Memcached -- a caching layer between the web servers and MySQL servers.

19

Web Database Application Architecture

- Typically, three tiers: database tier, logic tier, presentation tier
- In Web Database Application context,
  - Database Tier
    - comprises a back-end database or data store, comprising both data sets and the database management system software that manages and provides access to the data.
  - Logic Tier
    - A middle dynamic content processing and generation level application server
  - Presentation Tier
    - A front-end web server serving static content, and potentially some cached dynamic content. In web-based application, front end is the content rendered by the browser. The content may be static or generated dynamically.

20



21

Frequently-complained issues

- It takes too long to load my web page!
- More like than not, the culprit is the database back-end.
- Database performance matters

22

Database Performance

- Throughput: How much is the database doing?
  - The number of requests the database receives.
  - The number of transactions per second.
  - How many orders can the system take a second, how many web page requests can it service, etc.

23

Database Performance

- Execution time: How long does it take the database to do its job?
  - You don't just want to know how many requests the database received, but also how long the database spent on each request.

24

Database Performance

- Concurrency: How many jobs is the database doing at the same time?
  - The number of concurrent tasks changes the way the database's resources are used.
  - Concurrency can also affect latency, which includes not only the time it takes for the task to be completed (execution time) but also the time the task needs to wait before it's handled.

25

Database Performance

- Utilization: What percentage of the time was the database busy?
  - Utilization is a culmination of throughput, execution time, and concurrency to determine how often the database was available—or alternatively, how often the database was too busy to respond to a request.

26

Database Performance

- The primary focus of this course is how to address Database Performance issues.
  - Database Optimisation (Indexing)
  - Database Optimisation (Partitioning and Clustering)
  - Query Optimisation
  - Triggers
  - Stored Procedures

27

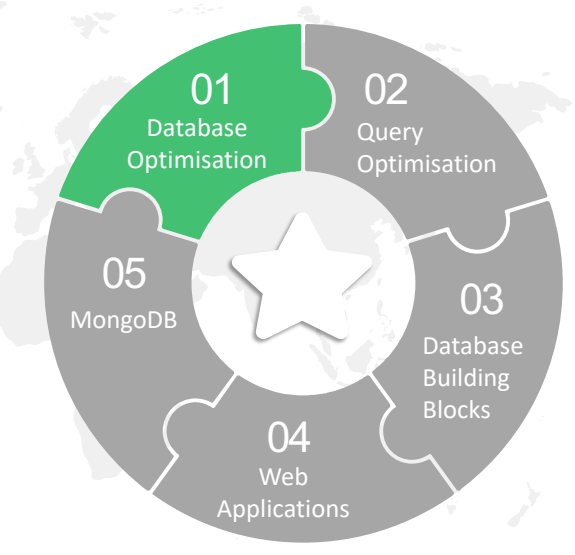
What will you learn in this course?

28

1

Database Optimisation.

In the beginning of the course, we explore ways of improving database performance by making changes to how the database is configured and built. For example, we explore the impact of partitioning an extremely large table, impact of having various indexes and how to configure physical storage characteristics to improve database performance.



2

Query Optimisation.

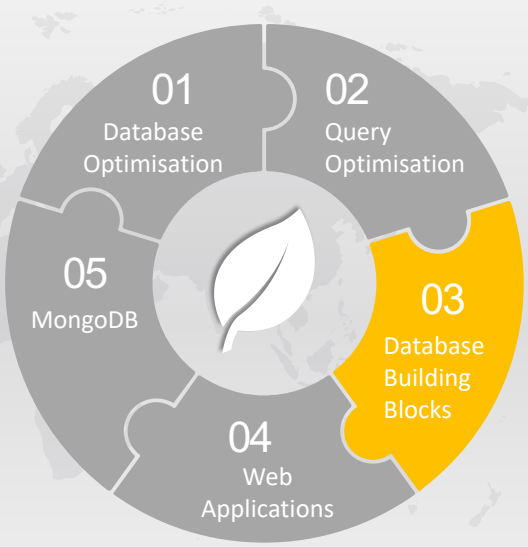
There are many different ways that you can write a query. In most of the time, in-built query optimiser takes care of converting your non-optimal query into an optimized query. However, as a database programmer, there are ways that you can assist the query optimizer by writing better queries.

3

Database Building Blocks.

You know how to create tables to store data. However, in order to run queries efficiently sometimes we need other database auxiliary objects, such as triggers, views, materialized views, stored procedures, etc.

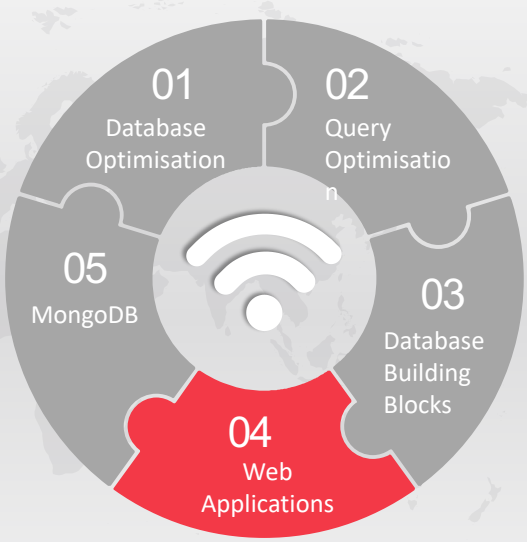
In the third part of the course we learn about them and spend a few weeks building them.



4

Web Applications.

Supplementing our lectures, we have a series of lab activities, where we explore how to build a web database application, using Oracle and PHP. These lab activities provide the skills required to complete your first assignment.



5

MongoDB.

In the last quarter of the semester, we move from relational database paradigm and focus on NoSQL approach. We learn the basic concepts as well as how to use MongoDB to store non-structured data. Some of the skills you build will be used in your second assignment.

6

Part 2: Database architecture and improving storage characteristics

29

Database architecture

- The database storage architecture can be described in terms of logical and physical structures.
- The advantage of separating the logical and physical structure is that the physical storage structure can be changed without affecting the logical structure.

30

Logical Structure

- The logical storage is defined as a hierarchical structures.
  - Tablespaces
  - Segments
  - Extents
  - Blocks

Logical Structure - Tablespaces

- Tablespaces
  - A database consists of one or more logical portions called as 'Tablespaces'. A tablespace is a logical grouping of related data.
  - Each database has at least one Tablespace called SYSTEM Tablespace. Data Dictionary is stored in this tablespace. User data are not supposed to be stored here.
  - You should create another tablespace, physically stored in a different location (disk, partition, remotely-mounted, clustered, etc) to store the database.

Logical Structure - Tablespaces

- Creating a Tablespace

```
SQL> CREATE TABLESPACE users
     DATAFILE '/dbhome/oradata/inventory/oltp01.dbf'
     SIZE 500M;
Tablespace created.
SQL>
```

The above SQL command will create a tablespace and a physical storage (a file) is assigned to the logical structure.

When this space is filled up, you can attach more physical file space, as follows.

```
SQL> ALTER TABLESPACE users
     ADD DATAFILE
             '/dbhome/oradata/inventory/oltp02.dbf'
     SIZE 1000M;
```


Listing Tablespaces and default physical parameters

To list the names, sizes, and associated tablespaces of a database, enter the following query on the `DBA_DATA_FILES` view

This is a view in the Data Dictionary. We discuss about them later

```
SELECT  FILE_NAME, BLOCKS, TABLESPACE_NAME
   FROM DBA_DATA_FILES;

FILE_NAME                                    BLOCKS  TABLESPACE_NAME
-----------                                ----------  ---------------
/U02/ORACLE/IDDB3/DBF/RBS01.DBF                1536  RBS
/U02/ORACLE/IDDB3/DBF/SYSTEM01.DBF             6586  SYSTEM
/U02/ORACLE/IDDB3/DBF/TEMP01.DBF               6400  TEMP
/U02/ORACLE/IDDB3/DBF/TESTTBS01.DBF            6400  TESTTBS
/U02/ORACLE/IDDB3/DBF/USERS01.DBF               384  USERS
```

More information:
http://download.oracle.com/docs/cd/B28359_01/server.111/b28310/tspaces014.htm

Why TABLESPACES?

- Tablespaces make it easier to allocate space quotas to various users in the database.
- Tablespaces enable you to perform partial backups and recoveries based on the tablespace as a unit.
- Because a large object like a data warehouse, partitioned table can be spread over several tablespaces, you can increase performance by spanning the tablespace over several disks and controllers.
- You can take a tablespace offline without having to bring down the entire database.
- Tablespaces are an easy way to allocate database space.
- You can import or export specific application data by using the import and export utilities at the tablespace level.

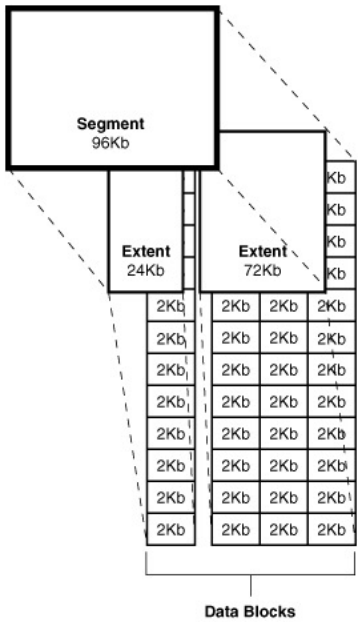
Logical Structure - Segments

- The space in a *tablespace* is divided into units called **segments**.
- There are four types of segments: Data, Index, Temporary, and Rollback.
- Each segment is used for their designated purpose.

Logical Structure - Extents & blocks

- Below the level of segment, the next level of granularity is called **an extent**.
- A segment can have a large number of extents.
- Extents help the DBMS to manage the growth of the database efficiently.
- Each extent consists of a set of contiguous database **blocks**.
- A database block is the lowest level of granularity at which Oracle performs disk I/O.
- A database block does not have to be the same size as a operating system block, but should be a multiple thereof.

Logical Structure



Logical Structure - Extents & blocks

- When a table is created, you can define its storage parameters, in terms of allocation of ***extents***.

```
SQL> CREATE TABLE dept
                 (deptno NUMBER(2),
                  deptname VARCHAR(50),
                  location VARCHAR(100))
           STORAGE (INITIAL 100K
                    NEXT 50K
                    PCTINCREASE 25);
```

100k	50k	62.5k	78.125k

Logical Structure - Extents & blocks

- You may locate a table in a chosen tablespace, by specifying the tablespace name.

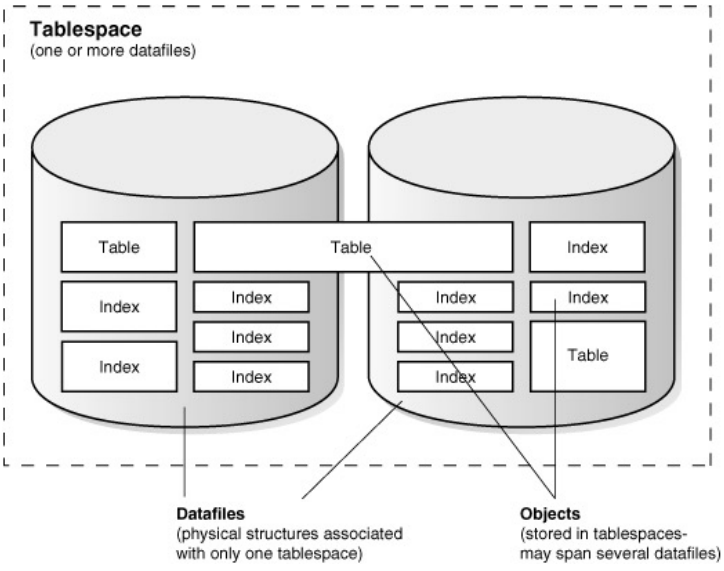
```
SQL> CREATE TABLE dept
                 (deptno NUMBER(2),
                  deptname VARCHAR(50),
                  location VARCHAR(100))
           TABLESPACE users
           STORAGE (INITIAL 100K
                    NEXT 50K
                    PCTINCREASE 25);
```

Physical Structure

- **Datafiles** are the operating system files that hold database data. The data is written to these files in an database-proprietary format that cannot be read by programs other than an DBMS.
- When tablespaces (logical structures) are created a mapping is done between them and datafiles (physical structures).
- E.g.

```
SQL> CREATE TABLESPACE users
     DATAFILE '/dbhome/oradata/inventory/oltp01.dbf'
     SIZE 500M;
Tablespace created.
SQL>
```


Physical Structure - Datafiles



Week 2 reading pack

Complete Week 2 Tute/Lab sessions

Next week, we explore database optimisation using indexing.