Oracle Database Design - 2024

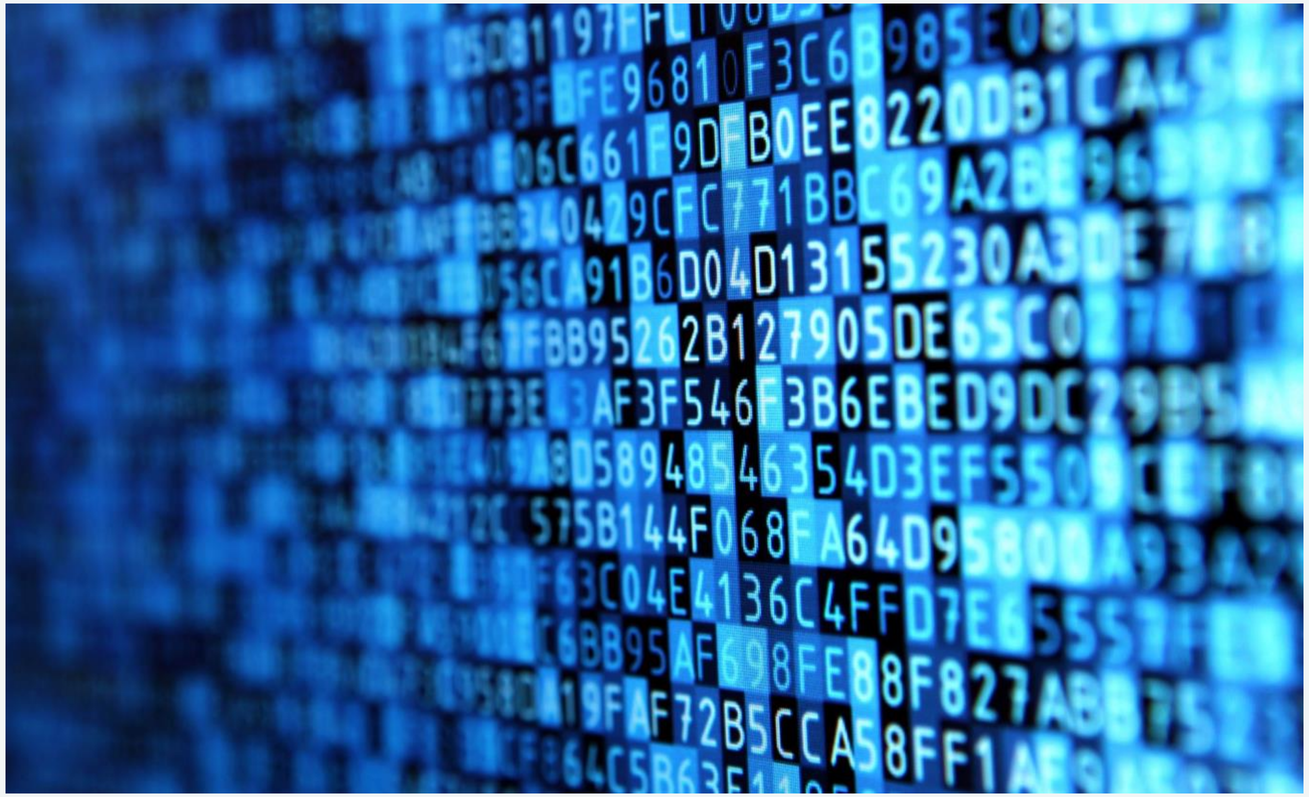
KRYSTIAN WOJTKIEWICZ

How to successfully finish the course?

1. There are 7 lectures but only 6 packed with knowledge
    - Introduction and rules outline
    - Getting into the topic
    - Exam
2. Laboratories
    - 4 lists
    - Every list has a deadline by which it must be completed and handed out
    - If you fail to keep the deadline the points for the list are halved each week
    - If you get enough points, the grade from laboratory might be copied to exam ☺
3. General rule
    - If you cheat, you fail

Lecture plan

1. Preliminary information.
2. SQL language - Oracle dialect.
3. PL/SQL language.
4. Object-oriented extensions of the Oracle database.
5. Bulk binding



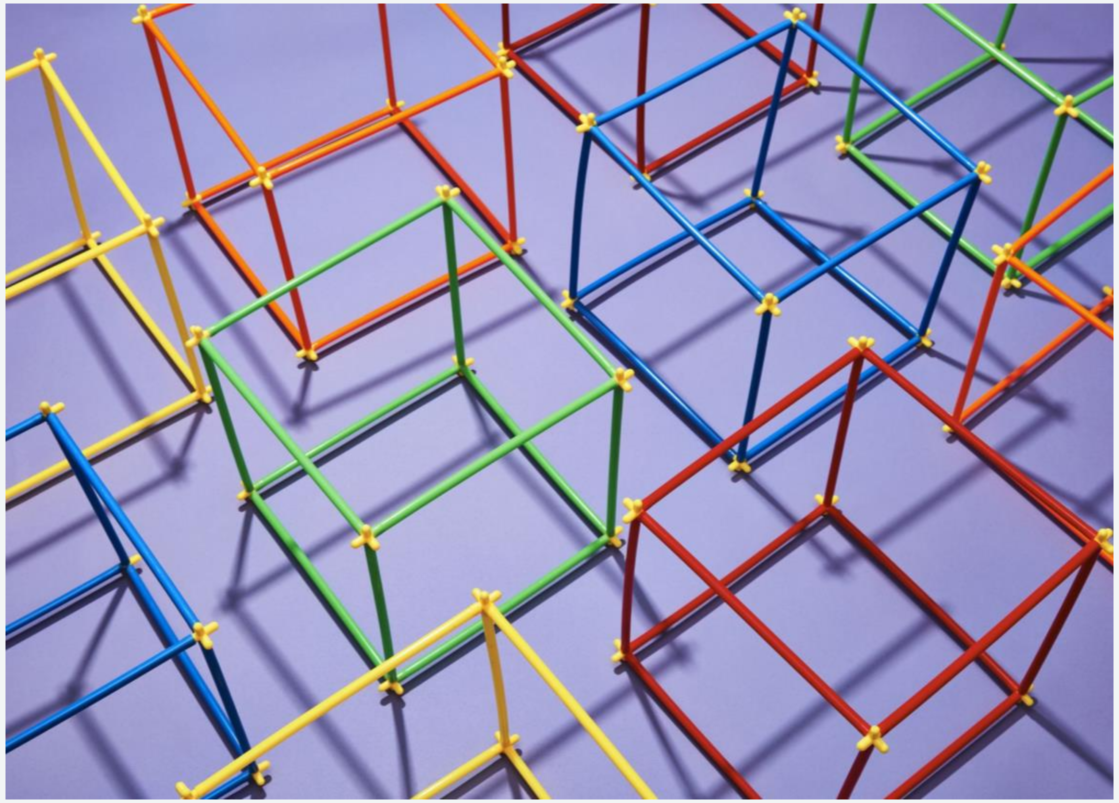
Preliminary information

Our civilization is a data civilization. Scientific theories are built based on collected **physical data.** Business **data** serves as the basis for making all decisions, and **data** is finally a value in itself - it is collected and searched. Due to the growing "resources" of **data**, a computer has become a natural tool for their collection and processing. Computer systems in which **data** is collected and managed are called **Database Management Systems (DBMS)**.

Among these systems, one of the most important is the system offered by Oracle.

Preliminary information

The Oracle database is a collection of data stored in files. It has its logical structure (links between collected data) and physical structure (set of physical objects in which data is collected). The Oracle database (instance) implementation consists of a memory area called the System Global Area (SGA) and background processes that communicate with the SGA area and database files.



The following types can be distinguished:

- Relational tables,
- Object-relational tables,
- Tables with an index structure,
- External tables,
- Partitioned tables
- Materialized Perspectives,
- Temporary tables,
- Clusters
- Removed tables.

The basic structures (objects) stored in the Oracle database are **tables**.

Preliminary information

Relational tables contain data entered and processed by the user.

Object-relational tables have data about user-defined types for which the inheritance mechanism can be used.

Tables with an **index structure** contain data stored in the index structure.

External tables are used to access extensive external data without loading it into the database.

Partitioned tables divide their data into partitions that can be managed separately.

Materialized perspectives contain the resulting data replica.

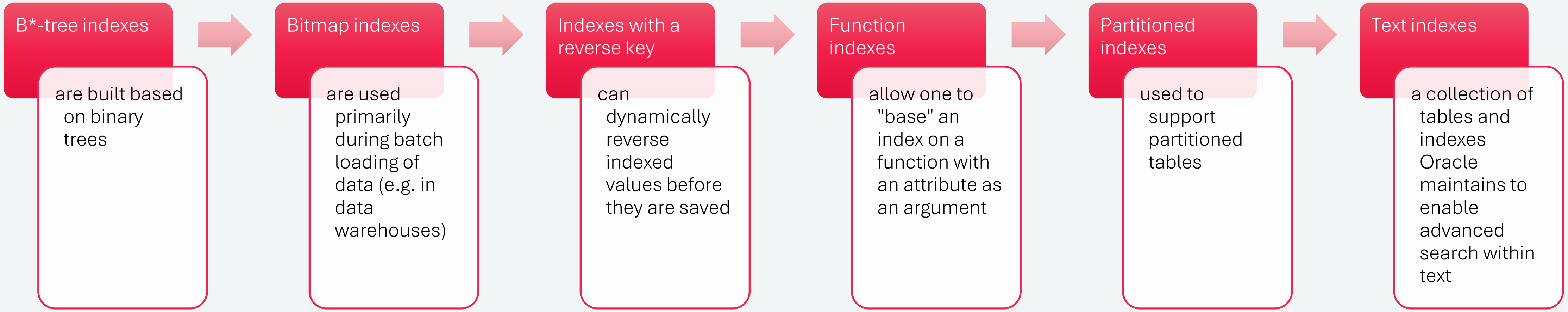
In **temporary tables**, each user can "view" only the rows entered by him.

One can save two tables in a **cluster structure**, to which queries are often directed together.

Deleted tables allow you to restore tables deleted using a command with special syntax quickly.

Preliminary information

To gain faster access to data stored in tables, Oracle uses structures (objects) stored in a database called indexes. The following types of indexes can be distinguished



- **B*-tree indexes**: are built based on binary trees
- **Bitmap indexes**: are used primarily during batch loading of data (e.g. in data warehouses)
- **Indexes with a reverse key**: can dynamically reverse indexed values before they are saved
- **Function indexes**: allow one to "base" an index on a function with an attribute as an argument
- **Partitioned indexes**: used to support partitioned tables
- **Text indexes**: a collection of tables and indexes Oracle maintains to enable advanced search within text

Preliminary information

In addition to relational tables and indexes, the Oracle system uses many other structures, such as **views**, **procedures**, **functions**, **triggers**, **packages**, **snapshots**, and **users**. *Most of these structures will be described in this lecture.*

The Oracle system has its dictionary, i.e., a database in which the data about the database (so-called metadata) are stored. These can be **user data**, **permissions**, **database object definitions**, **restrictions**, etc.

Database objects that require physical space in permanent memory in Oracle obtain them as part of the table space. The table space consists of one or more files. Each database object can be saved in one file or divided and saved in many files.

Preliminary information

- SQL language
- PL/SQL
- Dynamic SQL
- SQL*Plus
- Java and JDBC
- XML
- Object-oriented SQL and PL / SQL
- Data Pump
- SQL*Loader
- External programs and procedures
- UTL_MAIL

Description of the reality

After many years of independence, cats of both sexes hunted in the Wólka Mała village decided to organize themselves. So, a herd was created, commanded by the most excellent mouse hunter with the nickname Tiger. As part of the herd, under the leadership of the Tiger, an informal hierarchy of cats formed naturally. Each cat knew which other cat it was subordinated to. The herd was additionally administratively divided into several, having a unique number and name, of the bands, each commanded by an outstanding mouse hunter nominated by the Tiger. Each band was given an independent area where their cats could organize their hunts. Due to their high position, the Tiger and crew members had the privilege of hunting in the entire area controlled by the herd. For identification purposes, each cat was required to choose a unique nickname. The cat should also have a name. It was agreed that a member of the herd would be rewarded with a ration of mice every month for his contribution to maintaining the entire pack. This ration will be adequate to the function performed in the feline community. This function will define the lower and upper limits of the mice ratio. Regardless of the mice ration size, the herd leader, for special merits, will be able to grant the cat, at his own discretion, additional mice ration. Cats hunted happily in their assigned areas, but occasionally, there were incidents with representatives of other breeds. The participants of the incidents, identified by their names, automatically became the personal enemies of the wronged cats. Their degree of hostility and their species were carefully noted. All these events were described as a warning to cats and infamy for enemies (mandatory with their date). Assuming, however, that a real hunter can avoid known enemies, only the first cat incident with a specific enemy was recorded. Over time, cats noticed that handing bribes to enemies can reduce their alertness. For this reason, the sort of bribe preferred by each enemy was noted.

Basic structures

Cats(nickname, name, gender, in_herd_since, mice_ration, mice_extra, #function, #chief, #band_no)

Bands(band_no, name, site, #band_chief)

Functions(function, min_mice, max_mice)

Enemies(enemy_name, hostility_degree, species, bribe)

Incidents(#nickname, #enemy_name, incident_date, incident_desc)

underline means primary key

# means foreign key

The foreign key `function` in the `Cats` relation binds it with the `Functions` relation, the foreign key `band_no` bids it with the `Bands` relation, and the foreign key `chief` binds it with herself (points to the cat's superior). The foreign key `band_chief` in the `Bands` relation binds her with `Cats` relation. The foreign keys `nickname` and `enemy_name` in the `Incidents` relation bind her with the `Cats` and `Enemies` relations.