**Week 13, Lecture 26 Physical DB Design**

The Entity Relationship Diagram (ERD) is the conceptual data model of a database system. It visualizes the requirements of the database system, and specifies the overall structure of the data that must be managed in an organization or enterprise.
The ERD is independent of the specific database technology (such as relational database) to be used for the database system.
**Conceptual database design** is developing ERD for a database system and is also called database analysis.

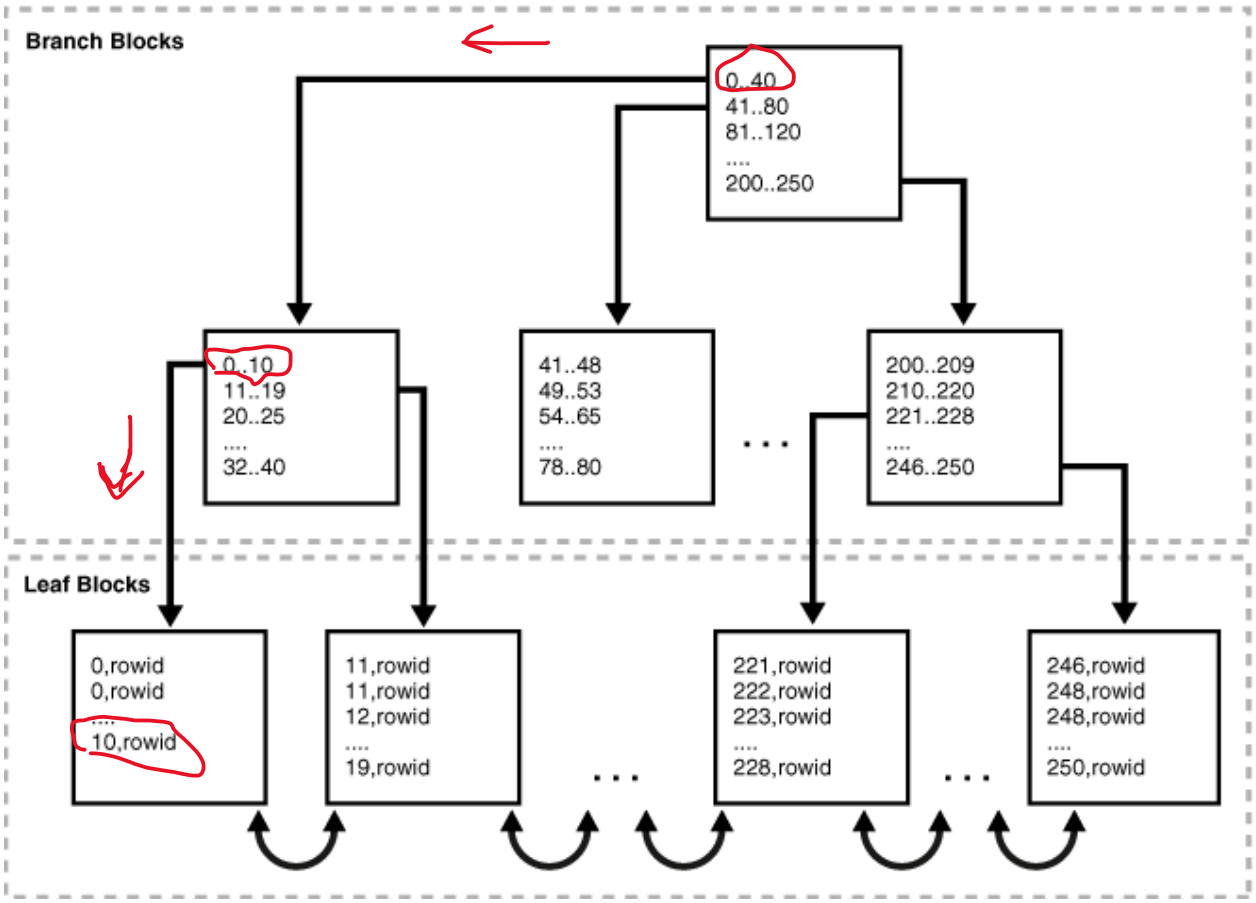
The Relational Schema is the logical data model of a database system for the chosen specific database technology: relational database.
**Logical Database Design** is developing the relational schemas for a database system.

**Physical Database Design** develops the specification of how the data in the relational schema is stored on the secondary memory (typically disk) and processed. The primary goal of physical database design is data processing efficiency.

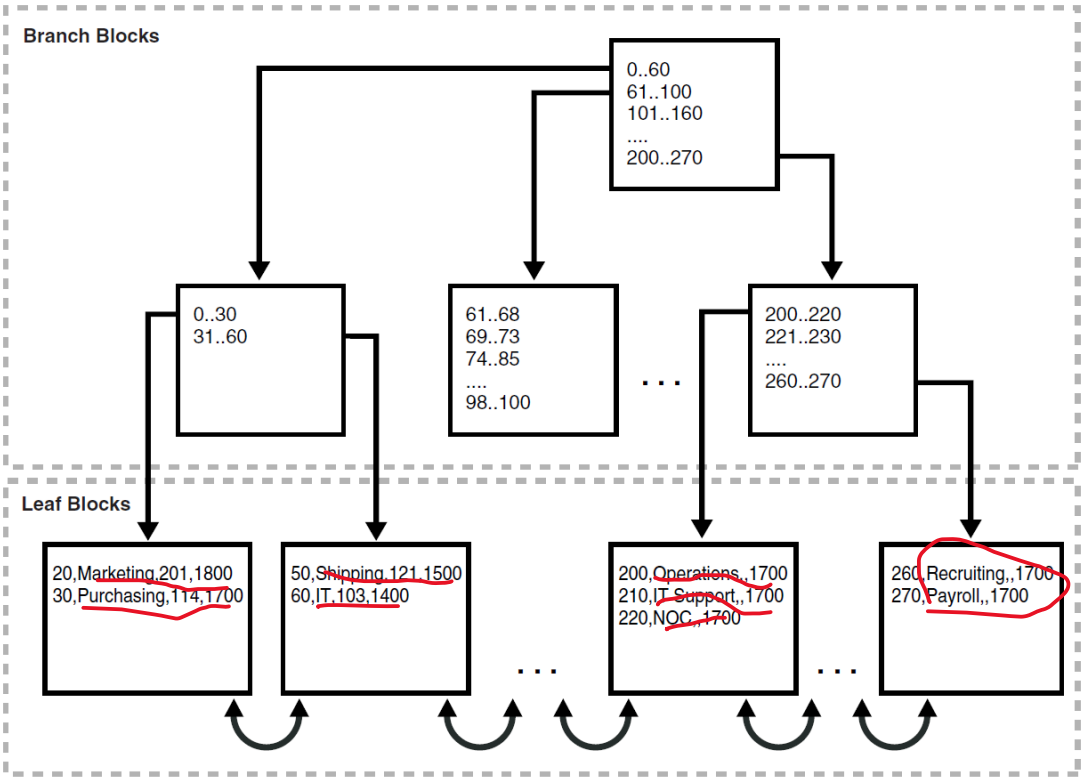
Some issues related to physical database design:

- Design table columns
  - Choose Data Type
    - Common data types in Oracle:
      - CHAR(n)
      - VARCHAR2(n)
      - NUMBER(n, p)
      - DATE
  - Control data integrity at physical storage
    - Default values, null values, range of values, legitimate values in foreign key columns, etc
- Indexing
  - To speed up query processing, indexes can be created for one or more columns in a table. They are used just like book indexes.
  - An index is a collection of index entries.
  - Each index entry is like a book index entry. Given an index for the empid column, an example index entry is: (10, location of row containing employee 10 such as rowid in Oracle). Empid is the index key. 10 is the index key value.
  - Index Structures
    - **B-Tree – most common type**
      - Each index entry corresponds to one table row.
      - Index Entries are stored in a **balanced tree** structure (see the figure in next page)
      - **Leaf blocks** at the bottom level of the tree store index entries.
      - **Branch blocks** guide the search in the correct direction based on the index key value range where the target key value falls. Branch blocks contain index key values and links to lower level blocks (branch blocks or leaf blocks). 0..40 means the range of index key values from 0 to 40.
      - Each exact search follows one path from the top branch block (called tree root) to a leaf block. For example an exact search for index key value 10 will take the path shown in RED color.
      - Data stored in both Branch blocks and Leaf Blocks are sorted based on the index key values.

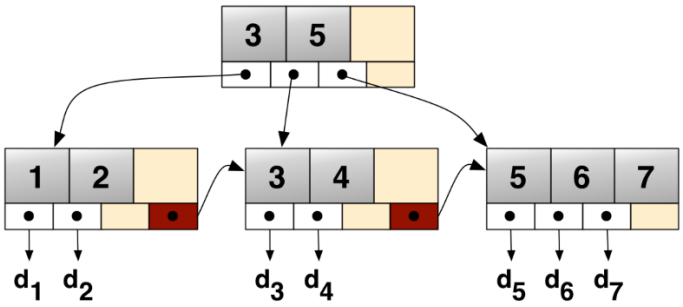
- Leaf blocks are linked in both directions for fast range queries such as locating any index key values smaller than 20.
- Balanced means that all leaf blocks are on the same level, resulting in faster queries.




- **B+Tree**
  - In Oracle, it is called Index Organized Table.
  - It's a variant of B-Tree. The only difference between B+Tree and B-Tree is that leaf blocks store the entire table row, instead of just the index key and the rowid for each table row. So the index is the table. The table is also the index. As show below. This shows an employee table stored as an index organized table (a B+ tree). The index key is the primary key empid. The differences between this B+ Tree and the B-Tree are marked in RED.



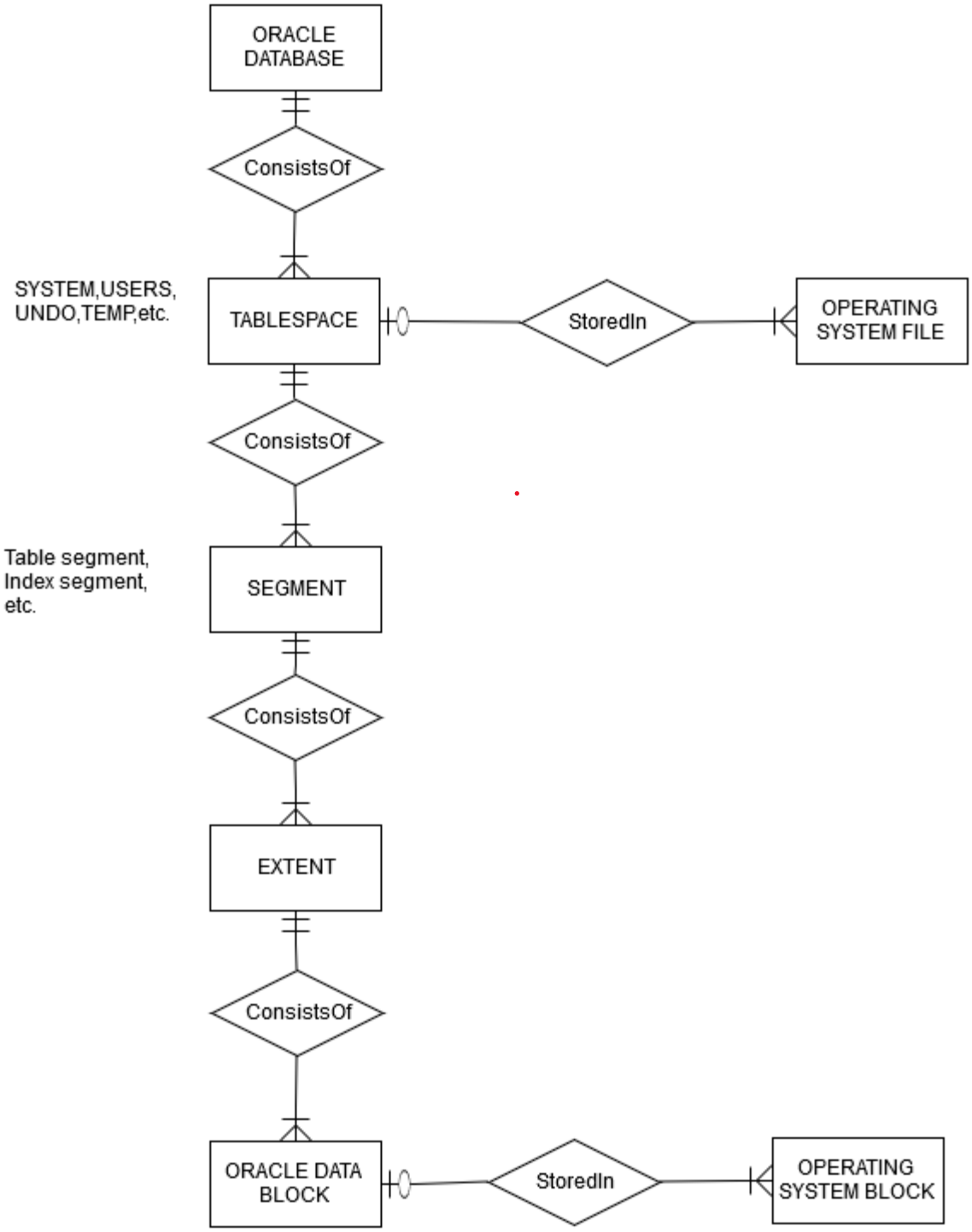

- Here is another example of B+ Tree. The numbers refer to the index key values such as employee ids. d1, d2, d3, d4, etc refer to the other data values in a table row containing the index key values 1, 2, 3, 4 respectively.




- **BitMap Index**
  - In a B-Tree index, one index entry corresponds to one table row.
  - In a BitMap index, one index entry corresponds to more than one row. An index entry is like:
    - (10, 010101) meaning index key value 10, the bit string means three table rows contains the key value 10: Row2, Row4, Row6. Each bit in the bit string is mapped (via a function) to the rowid of that row.

- **When to use Index (**some general rules)
  - Index a table column **when most operations on the table are queries,** but not data modifications. Indices speed up the data retrievals at the cost of data modifications (insertion, deletion, update)
  - Index a table **column if a query on the indexed column is very selective,** i.e. the query only returns a very small percentage of rows in the table. Otherwise, the query would run faster with a full table scan (meaning do a linear search on the entire table without using the index)

        - Index a table column **if the indexed columns are frequently queried**. Otherwise the cost of maintaining the index is not justified.
        - **Use B-Tree index for high-cardinality columns, use BitMap index for low-cardinality** columns (such as marital status, gender, etc). The **cardinality of a table column** refers to the **percentage of distinct values in the column.** Note: Oracle automatically creates an index for each primary key and each unique key.
- Manage storage space
    - Details are specific to a particular RDBMS product.
    - Example: Oracle storage hierarchy
        - An oracle database consists of one or more tablespaces.
        - **Tablespace**
            - A logical storage unit to store one or more tables, or other database objects like index.
            - Example Tablespaces in Oracle 12c
                - SYSTEM, SYSAUX: store definitions of database objects: tables, indexes, etc.
                - TEMP: for temporary work space, such as intermediate result during sorting.
                - UNDO: for rollback operations
                - USERS: for regular user data such as data in table employee.
            - Each tablespace can be stored in one or more operating system files.
        - **Segment**
            - Each tablespace is divided into a number of smaller units called Segments.
            - Each segment consists of data in a table, an index, or a partition.
        - **Extent**
            - Each segment is divided into a number of smaller units called Extents.
            - Each extent consists of a number of contiguous Oracle data blocks.
        - **Oracle Data block**
            - Smallest storage unit in Oracle
            - Each Oracle data block consists of one or more operating system blocks.
        - The ER Diagram for Oracle Storage Hierarchy is shown in next page.
- **Partitioning Data**
    - Split the data **in a table into multiple segments in** order to speed up the query processing
    - Example:
        - List partitioning
            - **customers table may be partitioned and split into multiple segments based on region** column. All rows with the same region are stored in one table segment.
        - Range partitioning
            - **transactions table may be partitioned and split into multiple segments based on date-time** column. All rows within a time period are stored in one table segment.


ORACLE DATABASE
ConsistsOf
SYSTEM,USERS, UNDO,TEMP,etc.
TABLESPACE
StoredIn
OPERATING SYSTEM FILE
ConsistsOf
Table segment, Index segment, etc.
SEGMENT
ConsistsOf
EXTENT
ConsistsOf
ORACLE DATA BLOCK
StoredIn
OPERATING SYSTEM BLOCK