IT2201 / IT2601 / IT2564 / IT2621 / IT2521 / IT2323

Database Management Systems

Unit 7
Physical Database Design

1

Topic Objectives

- At the end of this topic, you should be able to:
  - Describe the purpose and process of physical database design.
  - Apply the methodology in physical database design.

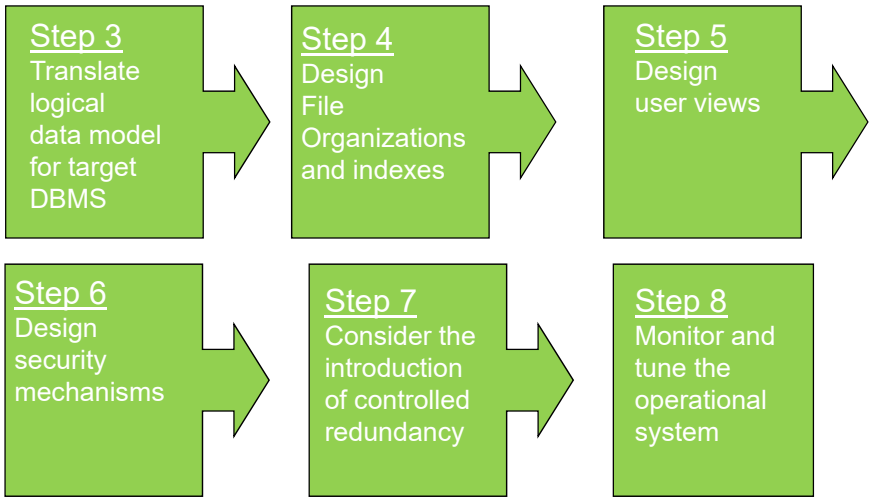
2

Logical vs. Physical Database Design

- Logical Database Design (What)
  - The process of constructing a model of the information used in an enterprise;
  - It is independent of the functionality provided by the target DBMS, but dependent on the target data model (e.g. in our course, 'relational' data model is adopted).
- Physical Database Design (How)
  - The process of producing a description of the implementation of the database on secondary storage;
  - It requires the physical database designer to know the functionality of the target DBMS.

3

Overview of Physical Database Design



4

Note: First 2 steps in logical database design

Step 3: Translate logical data model for target DBMS

- Objective
  - To produce a relational database schema that can be implemented in the target DBMS from the global logical data model.
  - Steps
    - 3.1 Design base relations
    - 3.2 Design representation of derived data
    - 3.3 Design general constraints

5

Step 3.1: Design Base Relations

- Objective
  - To decide how to represent base relations identified in logical data model in the target DBMS.
- Tasks
  - Collate and assimilate information about the relations from the logical data model and data dictionary (produced during the logical database design) :
    - name of the relation
    - a list of simple attributes, any derived attributes
    - the primary key, alternate keys and foreign keys
    - referential integrity constraints
  - Decide how to implement the base relations using target DBMS
  - Document design of base relations.

6

Documentation for the Base Relation

Consider the following relation in the logical data model:
Customer (custNum, cust_name, cust_addr)

The base relation is designed as follows:

Table Name: Customer
Table Description: This table stores customer details.

Column No.	Column Name	Column Description	Column Format	Key Type	Integrity Constraint
1.	custNum	Customer Number	N5	#	Not Null
2.	custName	Customer Name	C40		Not Null
3.	custAddr	Customer Address	C100		Not Null

Column Format:
c: character
n: number
d: date

Key Type:
#: primary key
f: foreign key
a: alternate key

Index: Nil

7

Step 3.2: Design Representation of Derived Data

- Objective
  - To decide how to represent any derived data present in the logical data model in the target DBMS.
- Task
  - Examine logical data model and data dictionary, and produce list of all derived attributes.
  - Derived attribute can be stored in database or calculated every time it is needed.
    - Consider additional cost to store the derived data vs cost to calculate it each time it is required;
    - Less expensive option is chosen subject to performance constraints.
  - Document design of derived data.

8

Customer Relation with Derived Attribute

Customer

Derive attribute

custNo	custName	custAddr	Acc_ balance
27889	John Little	33 High St	12,000
12008	Robinson	2 Queen St	28,000
23665	Johnson	11 Steven Rd	21,000

Acc_bal = ∑total order amt - ∑total payment amt

9

Step 3.3: Design general constraints

- Objective
  - To design the general constraints for the target DBMS.
- Task
  - Consider the remaining general constraints. Eg prevent a customer from owing debt > 100,000. We can define this constraint into the SQL Create Table as follows:

    CONSTRAINT debt_limit
    CHECK (value <= 100,000)
  - Document design of general constraints.

10

Step 4: Design File organizations and indexes

- Objective
  - To determine the optimal file organizations to store the base relations and the indexes that are required to achieve **acceptable performance**.
  - Steps
    - 4.1 Analyze transactions
    - 4.2 Choose file organizations
    - 4.3 Choose indexes
    - 4.4 Estimate disk space requirements

11

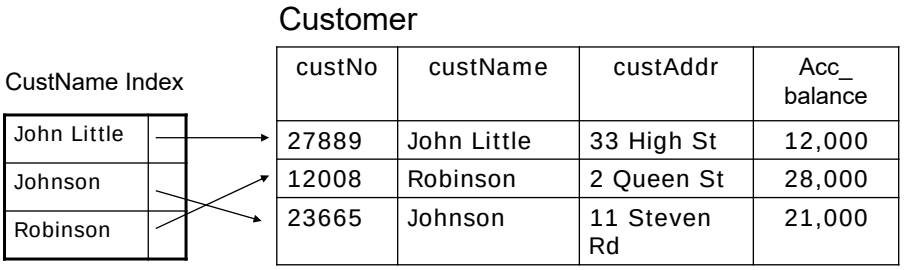
Indexes

- What is an Index ?
  - Indexes are data structures which points to the data rows.
  - It is used for locating and retrieving of data more quickly.
  - It helps to avoid the need to scan the table sequentially for a particular record.
  - Similar to an index at the end of a textbook.

12

Indexes

- An index contains records consisting of the key value and the address of the records in the data file.
- The below figure illustrates an index created on the customer name column.



13

Types of Index

- There are different types of index, the main ones being:
  - Primary index
    - The data file is sequentially ordered by an ordering key field, and the index is built on the ordering key field, which is guarantee to have a unique value in each record.
  - Clustering index
    - The data is sequentially ordered on a non-key field, and the indexing is built on this non-key field, so that there can be more than one record corresponding to a value of the indexing field.
  - Secondary index
    - An index that is defined on a non-ordering field of the data file.

14

Types of Indexes

Remember that Indexes are built on columns:

Type of Index	Is this column a key ?	Data sorted on this column ?
Primary Index	Yes	Yes
Clustering Index	No	Yes
Secondary Index	No	No

15

Review question 1

- In the following Student relation, the primary key is adminNo and the data file is ordered in adminNo sequence.
  - Student (AdminNo, Name, NRIC, address)
- We created two indexes on this relation: one is adminNo and the other is NRIC. Identify type of these two indexes.
  - Ans:
- If we order the file in Name sequence and create an index on Name. what is this type of index?
  - Ans:

16

Choose Indexes

- Objective
  - To determine whether adding indexes will improve the performance of the system.
- Purpose
  - Secondary indexes provide a mechanism for specifying an additional key for a base relation that can be used to retrieve data more efficiently.
- Considerations
  - Overhead involved in the maintenance and use of secondary indexes vs performance improvement gained

17

Specifying Indexes

- Task
  - Create index using the SQL statement
  - For example, to create a primary index on the Customer relation based on the custNo attribute:
    - CREATE UNIQUE INDEX custNoInx ON Customer (custNo);
  - Create a secondary index on the Customer relation based on the custName attribute:
    - CREATE INDEX custNameInx ON Customer(custName);
  - Document choice of secondary indexes.

18

Guidelines for choosing secondary indexes

- When to create an Index ?
  - Index the primary key of a relation if it is not a key of the file organization.
  - Add secondary index to a foreign key if it is frequently accessed.
  - Add secondary index on columns that are involved in:
    - selection or join criteria
    - ORDER BY
    - GROUP BY
- When not to create an index ?
  - Do not index small tables.
  - Avoid indexing a column or table that is frequently updated.
  - Avoid indexing a column if the query will retrieve a significant proportion (e.g. 25%) of the rows in the table.
  - Avoid indexing columns that consist of long character strings.

19

Step 5: Design User Views

- Objective
  - To design the user views that were identified during the Requirements Collection and Analysis stage of the relational database application lifecycle.
- Advantages of user views :
  - Data independence
  - Reduced complexity
- Task
  - Create views using CREATE VIEW SQL statement.
  - Document design of user views.
- For example, to restrict users to access to a subset of staff particulars, a view allStaff is created as follows:

```
CREATE VIEW allStaff
AS SELECT  staffNo, fname, lname, branchNo
        FROM  staff;
```

20

Step 6: Design Security Measures

- Objective
  - To design the security measures for the database as specified by the users.
- 2 types of database security :
  - **System security** – covers access at system level, such as a user name and password.
  - **Data security** – covers access of database objects, such as relations and views, and the actions that users can have on the objects.
- Task
  - Design access rules using GRANT, REVOKE SQL statement
  - Document design of security measures.
- Example: Give users Personnel and Director the privileges SELECT and UPDATE on column salary of the Staff table.

```
GRANT SELECT, UPDATE (salary)
ON staff
TO Personnel, Director;
```

21

Step 7: Consider the Introduction of Controlled Redundancy

- Objective
  - To determine whether introducing redundancy in a controlled manner by relaxing the normalization rules will improve the performance of the system.
- The results of normalization is a logical database design that is structurally consistent and has minimal redundancy.
- A normalized database design does not provide maximum processing efficiency.
- In favor of performance, denormalization may be necessary.

22

Denormalization

- Denormalization refers to
  - Refinement to the relational schema (i.e. lessen the degree of normalization);
  - Combine two relations into one new relation;
  - Also called usage refinement.
- Factors to be considered
  - Denormalization makes implementation more complex.
  - Denormalization often sacrifices flexibility.
  - Denormalization may speed up retrievals but it slows down updates.

23

Denormalization

- When to Denormalize
  - Performance is unsatisfactory.
  - Relations have low update rate and a very high query rate.
- Where to Denormalize
  - Consider derived data.
  - Consider duplicating attributes or joining relations together (to reduce the no. of joins).

24

Denormalization

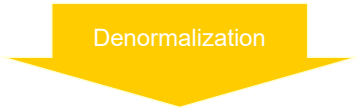
- Duplicating Attributes or Joining Relations
  - An Example

Branch (3NF)

BNO	Street	PCode	Phone	Fax

PostCode (3NF)

PCode	Area	City



Branch (2NF)

BNO	Street	Area	City	PCode	Phone	Fax

25

Denormalization

- Consider Denormalization in the following situations.
  - Combining one-to-one (1:1) relationships.
  - Duplicating nonkey attributes in one-to-many (1:M) relationships to reduce joins.
  - Duplicating foreign key attributes in one-to-many (1:M) relationships to reduce joins.
  - Duplicating attributes in many-to-many(M:M) relationships to reduce joins.
  - Introducing repeating groups.
  - Merging lookup tables with base relations.
  - Creating extract tables.

26

Step 7: Consider the Introduction of Controlled Redundancy (cont'd)

- Document introduction of redundancy
  - The introduction of redundancy should be fully documented, along with the reasons for introducing it.
  - Document the reasons for selecting one approach where many alternatives exist.
  - Update the logical data model to reflect any changes made as a result of denormalization.

27

Step 8: Monitor and Tune the Operational System

- Objective
  - To monitor the operational system and improve the performance of the system to correct inappropriate design decisions or reflect changing requirements.
- Many DBMSs provide the DBA with utilities to monitor the operation of the system and tune it.

28

Quiz

Across

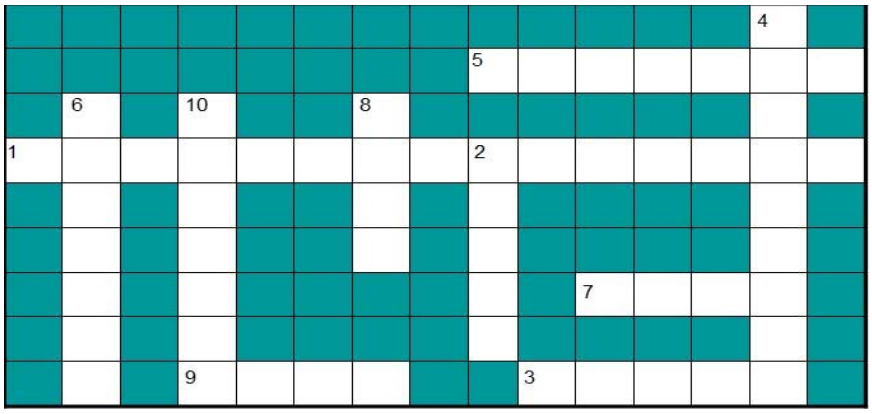
1. The process of transforming normalized relations into unnormalized physical record specifications.
3. Denormalize relations that have low update rate and high ____ rate.
5. Avoid indexing an attribute that is frequently _____.
7. _____ security covers access of database objects such as relations and views.
9. Avoid indexing columns that consists of _____ character strings.

Down

2. A table or other data structure used to determine the location of rows in a file that satisfies some condition.
4. An index that is defined on a non-ordering field of the data file.
6. An attribute that can be computed based on existing data is called _____ attribute.
8. The tuples of _____ relation are physically stored in the database.
10. _____Database Design is based on a specific data model, but independent of a particular DBMS.

29

Quiz



30

Summary

- Physical database design is the process of producing a description of the implementation of the database on secondary storage.
- It describes the base tables and indexes used to access this data effectively, and any associated integrity constraints and security restrictions.

31

Reference Materials

1. Database Systems, Connolly, Ch 18
2. Database Solutions, Connolly, Ch 12-16

32