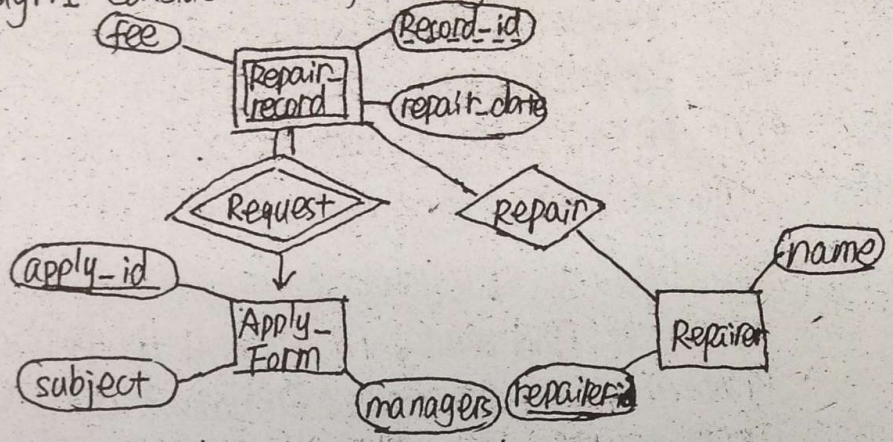
1. A data model is a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.
2. The three-level of data abstraction in database system includes: physical level, logic level and the view level.
3. The participation of an entry set E in a relationship set R is said to be full if every entity in E participate in relationship in at least one in R. If only some entities in E participate in relationship set R, the participation is said to be partial
4. A view is a virtual relation that is not part of the logical model, but is made visible to a user.
5. Integrity constrains guard against accidental damage to the database. The allowed integrity constrains in relational database include primary key, not null, default and unique predicate.
6. The ACID properties of transaction are: atomicity, consistence, isolation and durability.
7. An index entry consists of a search key and pointer to one more records in the data file.
8. Consider a B^+-tree of order n. if there are K search-key values in the file, the path from the root to the leaf node is no longer than $\lceil \log_{\lceil \frac{n}{2} \rceil} K \rceil$
9. The two most important heuristic rule is: (a) perform selections operations as early as possible; b) perform projections early.
10. Data items can be locked for a transaction in two modes: S mode and X mode in lock-based concurrency control scheme.
11. The immediate modification scheme allows database modifications to be output to the database while the transaction is still in the active state

1. Database design I: Consider the following E-R diagrams.



a) list the entity sets and their primary keys

b) construct appropriate relation schemas for the above ER diagrams.

c) Give a SQL DDL definition for the table exported from relationship set with appropriate data-type defined in standard SQL. Identify referen -integrity constrains that should hold, and include them in the DDL definition.

2. Database design II: Consider a relation schema R(A,B,C,D,E) and its functional dependencies. F={A→C, C→A, B→AC, D→AC} complete the following questions:

a) Compute (BD)+ ABCD

b) Compute the candidate keys for R. BE

c) Give a decomposition into BCNF of schema R.

3. Relational Algebra and SQL.

Consider a database with the schemas below:

SUPPLIER (sno, sname, scity)

PART (pno, pname, pcolor, weight)

PROJECT (jno, jname, jcity).

SPJ (sno, pno, jno, quantity).

Notes: The relation SUPPLIER describes the information about supplies, which has a primary key named sno. The relation PART describes the information about parts with pno as the primary key. The relation PROJECT describes the information about projects with jno as the primary key. The relation SPJ associate supplier with part and project, which denotes which supplier will supply parts for a project stored in PROJECT.

a) Give an expression in SQL to express the query: Find the pno of the lightest part.

b) Give an expression in SQL to express the query: Find the sname and scity of supplier that can supply all parts for project numbered as 'J1'.

c) Give an expression in SQL to express the query: List the amount of parts for each project in ascending order.

d) Give an expression in SQL to express the operation: delete the information about the supplier whose name is 'ITM'.

e) Give an expression in SQL to express the operation: add a new project located in 'shanghai' to the database. the project name is 'Sys'.

f) Give an expression in relational algebra to express the query: Find the sno of suppliers that supply red parts for project numbered as 'J1'

g) Give an expression in relational algebra to express the query :Q5: Find the JNOs of projects that doesn't use red parts produced by supplier from 'tianjin'.

4. Query Processing

a) Given the query "SELECT jno FROM supplier s, spj, part p WHERE s.name='sony' and p.color='red' and s.sno=spj.sno and spj.pno=p.pno". Draw the query expression tree for the above query without optimization.

b) Let M denote the number of page frames in the main-memory buffer, b_p and b_{spj} denote the number of blocks containing tuples of relation part and spj respectively. For a nature-join of p(PART) and spj(SPJ), estimate the number of block transferred. (For the above nature-join using a merge-join, and we don't know whether p and spj are orderly or not. Ignore the saving of final result.).