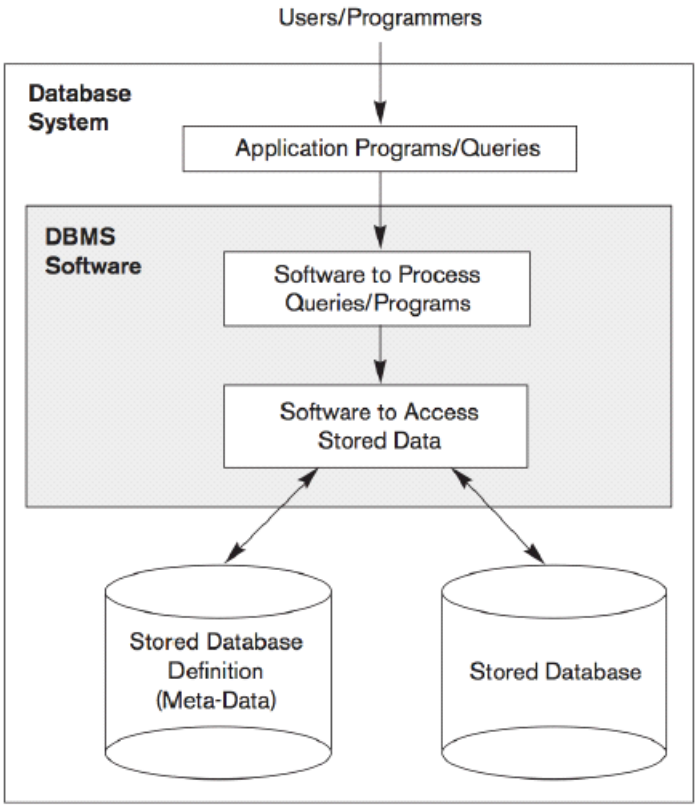
Review1. Database System

2019年5月27日 19:14

Database basic

- *Data: a collection of symbols recorded things, it represent some meaning (information) only if it's interpreted.*
  - *Information: factors (meaning, humans, assign to, data), concept / notation (holds only if there is a recipient, and is strongly linked to and is dependent on the values of the recipient).*
  - Data - interpretation, information - recipient.
- Encryption and decryption: information (plaintext) > encryption > ciphertext (data) > decryption > original plaintext (information).
- DBS (normally and strict, a body of information) and DBMS (software system).
  - *Database: a collection of logically related data stored structurally, organizationally, and together in a computing system.*
  - *Database management system: a software system running on a computing system that provides a systematic approach to manage data stored in a database and control access to the database.*
  - *Database application: a program that interacts with a database system at some point in its execution.*
  - ***DBS = DB + DBMS***
- DBS is a collection of data but not information.



- *File based systems: a collection of application programs (each program defines and manages its own data) that perform services for the end-users.*

    - Disadvantages: data redundancy (duplication) and inconsistency, separation and isolation, dependence (program-data dependence). Integrity, atomicity, concurrent-access, and security problems.
    - Intrinsic factors of the file-based approach:
        - Definition of data is embedded in the application programs, rather than being stored separately and independently.
        - No control over the access and manipulation of data beyond that imposed by the application programs.
- Conceptual layers of database implementation: user > application software (in terms of application) > database management system (database model) > actual database (actual organization).
    - DBMS as an interface, dichotomy between application programs and the DBMS.
- Functions of DBMS: DDL (data-definition language) and DML (data manipulation language) or query language.
- *Advantages of DBMS: control of data redundancy, data consistency, more information from the same amount of data, improved data integrity, sharing of data, improved security, enforcement of standards, economy of scale, balance of conflicting requirements, improved data accessibility and responsiveness, increased productivity, improved maintenance through data independence, increased concurrency, improved backup and recovery services.*
    - *Disadvantages of DBMS: complexity, size, cost of DBMS, additional hardware costs, cost of conversion, performance, greater impact of a failure, enterprise.*

Concepts and architecture

- Major purpose of a DBS: providing abstract view of the data.
    - Data abstraction levels: simplify user's interactions, retrieve efficiently, view level > logical level > physical level.
- Data abstraction:
    - Physical (internal) level: describes complex low-level data structures in details.
    - Logical (conceptual) level: describes what data is stored in DB, and what relationships exist among those data.
    - View (external) level: describes part of entire DB.
- Instances and schemas of DB: collection of data - instance, overall design - schema.
    - Schemas: sub-schema (external schema) - view, logical (conceptual) schema (most important) - DB design of logical level (entities, attributes, relationships, integrity constraints), physical (internal) schema - definition of stored records, methods of representation, data fields, indexes, storage structures used.
        - Physical data independence: program independent on physical schema.
- Architecture of DBS: DBTG two-level (schema, sub-schemas), ANSI three-level (external > conceptual > internal level > physical data).
    - DBMS: mapping between schemas (check consistency), two-stage mapping.

    - Data independent: logical and physical.
- Data model: structure, operation, constraint of data.
    - Relational model: tables (relations).
    - Record-based model, entity-relationship model, object-oriented data model, object-relational data model, semi-structure data model, network data model and hierarchical data model.
- Data language: DDL, DML, query language.
    - DML: procedural DML, declarative (nonprocedural) DML.
    - DDL: consistency and domain constraints, referential integrity, assertion, authorization (read, insert, update, delete).
- Database design: initial phase, conceptual-design phase, logical-design phase, physical-design phase.
    - Functional components of DBS: storage manager (authorization and integrity, transaction, file, buffer, data structures - files, dictionary, indices), query processor (DDL interpreter, DML compiler, query evaluation engine).
    - ACID: atomicity, consistency, isolation, durability.
    - Transaction: recovery manager, concurrency-control manager.
    - ***Query processor = query compiler + execution engine***
    - ***Transaction processor = logging and recovery manager + concurrency − control manager (scheduler)***
- Full-scale DBMS: data storage, retrieval, and update; user-accessible catalog; transaction support; concurrency control, recovery, authorization, integrity, utility services; data communication, independence.
- DB application architecture: client machines and server machines.
    - Two-tier and three-tier architecture.

Review2. Relational Database and the Relational Model

2019年5月27日 19:14

Notation of relation

- Ordered pair, Cartesian product, and Cartesian power.
- Binary relation: source (from-set) to target (to-set), domain, range.
  - Universal relation, empty relation, identity relation, inverse relation; reflexive relation, connected relation, transitive relation, symmetric relation, antisymmetric relation; compatible, pseudo order, equivalence, partial order, total (linear) order relation.
- Triple relation and n-ary relation.
- *Definition: relational database, relational database model, relational database schema, relational algebra, relational data model.*
  - Relational database: based on the relational model and uses a set of tables (relations) to represent both data and relationships among those data.
  - Relation (table), attribute (column), and row (tuple).
  - Relational instance, domain of attributes (atomic), null value, relation intension, relation extension (state), degree, cardinality.
- Keys: expressed in terms of their attributes.
  - Superkey, candidate key (non-uniqueness of candidate keys), primary key, foreign key (referencing relation, referential integrity constraint).
- Database schema and instance: relation schema, schema diagram, mathematical relations.
  - Database: a collection of logically related data stored structurally, organizationally, and together in a computing system.
- Constraints: integrity, domain, referential.
  - Entity integrity and base relation.
- View: a virtual or derived relation, may be dynamically derived from one or more base relations.

Review3. Relational Algebra and Relational Calculus

2019年5月27日 19:14

Overview of relational algebra

- Operation: usual set operations (union, difference, intersection), remove parts of a relation (selection, projection), basic (fundamental) relational operations (Cartesian product, rename), combine (Cartesian product / direct product, join).
  - Query: an expression of relational algebra.
- Expression:

  (1) $E1 \cup E2$, (2) $E1 - E2$, (3) $E1 \times E2$, (4) $\sigma_P(E1)$, where P is a predicate on attributes in E1, (5) $\Pi_S(E1)$, where S is a list consisting of some of the attributes in E1, and (6) $\rho_x(E1)$, where x is the new name for the result of E1.

  - As I'll skip this part, cause of you should already learned in the homework 1.
  - Join: natural join, theta join, outer join, semi join; joined tuple, dangling tuple.
  - Assignment, division, generalized-projection, aggregation operation.

Logic calculus

- Alphabet: connectives, quantifiers, logical constants, punctuation, individual variables, constants, functions, predicates.

  ***Connectives*: ¬ (negation), → (material implication), ∧ (conjunction), ∨ (disjunction), ↔ (equivalence)**

  ***Logical constants*: T and ⊥**

  ***Propositional variables* (letters): $V =_{df} \{p_1, p_2, ..., p_n, ..., \}$**

- Formulas (well-formed formulas): atomic formula, subformulas (immediate, improper), open formulas (without quantifiers).
- Terms: closed terms (no individual variable).
- Prediction: free occurrences, bound occurrences (sentence / closed formula).

Relational calculus

- *Tuple relational calculus, domain relational calculus and relational calculus query.*
- Tuple relational calculus: $\{t \mid P(t)\}$
  - *Atomic formulas and formulas.*
  - *Infinite results and domain.*
  - Safety: all values appear in the result are values from domain of P.
- Domain relational calculus: $\{\langle x_1, x_2, \cdots, x_n \rangle \mid P(x_1, x_2, \cdots, x_n)\}$
  - *Atomic formulas, formulas, query, infinite results, safety.*

Review4. SQL: Structured Query Language

2019年5月27日 19:14

I'll skip this part as I thought you should already learnt in homework.

Review5. Database Design and the Entity-Relationship (ER) Model

2019年5月27日 19:14

DBS design

- Five phases of DBS design:
  - Initial phase: specification of user requirements.
  - Second phase: conceptual-design phase, entity-relationship diagram.
  - Third phase: specification of functional requirements.
  - Fourth phase: logical-design phase, relation schema.
  - Fifth phase: physical-design phase.
- Major pitfalls: redundancy, incompleteness, inaccurateness.
- DSDLC (DBS development lifecycle): software crisis, feedback loops.
  - *SDLC (software development lifecycle).*
- DBS planning: management activities allow stages of DSDLC realized.
  - Mission statement, mission objectives, development of standards.
- DBS definition: boundaries of the DBS, user views, requirements collection and analysis, requirement specifications (techniques), identifying, and managing.
  - DBS design: bottom-up approach (top-down approach) - conceptual design, logical design, physical design; data modeling.

ER model

- Basic: entity sets, relationship sets, and attributes.
  - *Relationship instance, participation, role, recursive relationship set, descriptive attributes.*
  - Attributes: simple attributes, composite attributes.
    - *Single-valued attributes and multi-valued attributes, primitive attributes and derived attributes.*
- Entity sets: weak entity sets and strong entity sets, identifying (owner) entity set.
- ER notations.

Review6. Relational Database Design

2019年5月27日 19:14

Methodology

- Atomic domain: elements of the domain are considered to be indivisible units.
- Normal form: 1NF (atomic), 2NF (partially dependent), 3NF (prime attribute and transitively dependent), BCNF, 4NF.
- FD (functional dependencies): proposition logic basically, and functionally determine.
  - Trivial FDs: satisfied by all relations.
  - Armstrong's axioms: reflexivity, augmentation, transitivity.
    - Derived rules: union, decomposition, pseudo-transitivity rules.
- Relational decomposition: lossless and lossy decomposition.

Review7. Database Application Design and Development

2019年5月27日 19:14

Application architectures

- User interface: web application architectures.
  - Presentation or user interface layer.
  - The business-logic layer.
  - The data access layer.