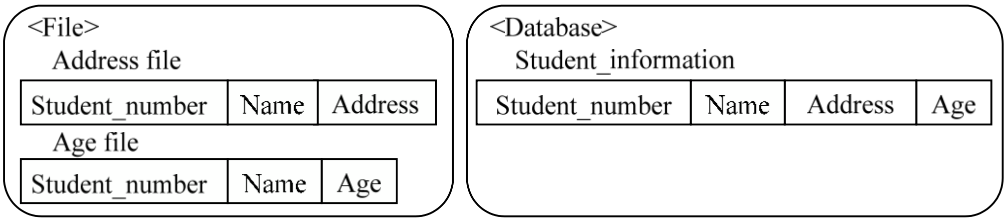
1 Outline of Database

Methods of handling data in computers include the method that uses files and the method that uses a database. A database is a method where data is consolidated and managed in an integrated manner. This section describes the concepts of a database and its operations.

1-1 Difference Between Database and File

Since the file is created for each program, there is a risk that the problem occurs – such as overlapping of data items and data inconsistency because of different contents of overlapping data items.

Therefore, as the integrated management of data, the concept of database becomes popular because it prevents overlapping of data and inconsistency of data.



<File>

Address file

Student_number	Name	Address

Age file

Student_number	Name	Age

<Database>

Student_information

Student_number	Name	Address	Age

- Overlapping of data items: Student number and name of the same student are recorded as overlapping in two files (address file and age file).
- Data inconsistency: When the name is modified in the address file but not in the age file, the student with the same student number has a different name.

Figure 4-1 File and database

The following functions are required for a database.

- Data sharing function
  It allows identical data to be provided to multiple users.
- Data independence function
  It ensures that data modification does not affect the program.
- Data maintenance function
  It ensures that data is always in the correct status (or there is no contradiction).
- Data fault countermeasures function
  It allows any fault of a database to be quickly addressed.
- Data security protection function

It ensures security of data by setting access rights, and so on.

1 - 2 Database Design

For using a database, it is necessary to analyze the data to be handled and design the database.

1-2-1 Data Model

Data model refers to modeling of various relations of data in the real world for handling them in computers. Data models are classified from the viewpoint of modeling.

- Conceptual data model
  It models relations of data without considering any specific database. It defines what kind of data is handled.
- Logical data model (external model)
  It logically models the relations of data considering a specific database. It defines interrelations of data in the database.
- Physical data model (internal model)
  It physically models the relations of data considering a specific database product. It defines the physical internal structure of a database such as data type.

Among these models, the following three data models are used as the logical data models considering a specific database.

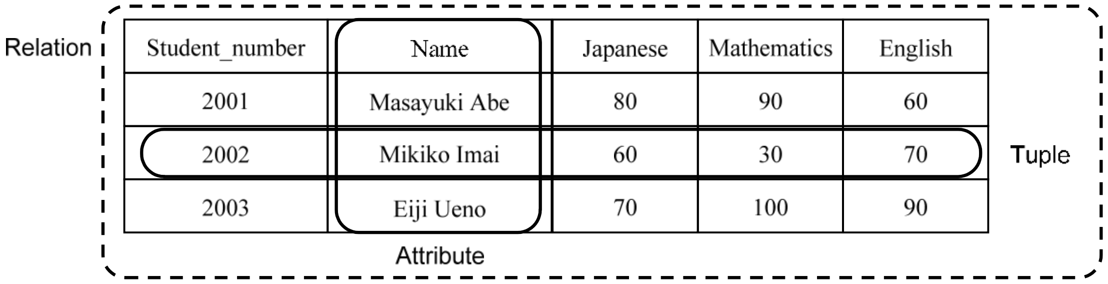
- Hierarchical model
  It is a data model that represents the relations of data as a hierarchical structure where parent and child are in a one-to-many relationship. It is implemented as HDB (Hierarchical DataBase).
- Network model
  It is a data model that represents the relations of data as a hierarchical structure where parent and child are in a many-to-many relationship. It is implemented as NDB (Network DataBase).
- Relational model
  It is a data model that represents the relations of data in a two-dimensional table. It is implemented as RDB (Relational DataBase).

The database, such as HDB (Hierarchical DataBase) or NDB (Network DataBase), is referred to as structured database. This is a type of database where the user recognizes logical layout of data.

In this textbook, the subsequent explanation is provided focusing on the RDB (Relational DataBase) that implements the relational model.

1-2-2 Relational Model

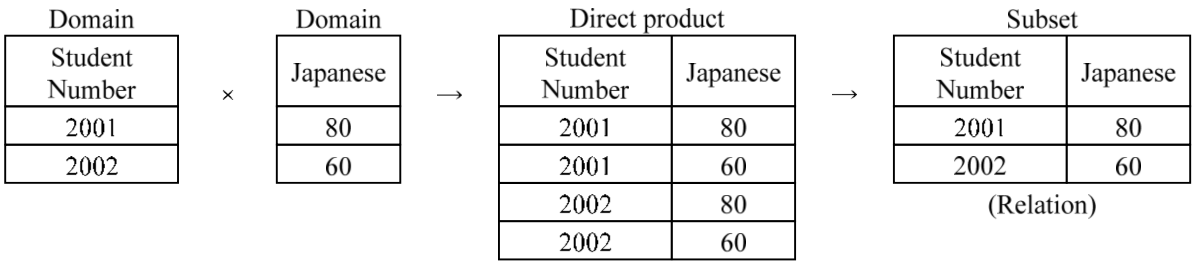
Relational model is a data model that represents the relations of data in a two-dimensional tabular form. The entire table is called "relation," rows and columns that constitute the table are called "tuples" and "attributes (fields)" respectively.



Student_number	Name	Japanese	Mathematics	English
2001	Masayuki Abe	80	90	60
2002	Mikiko Imai	60	30	70
2003	Eiji Ueno	70	100	90

Figure 4-2 Relational model

In Figure 4-2, values that are actually recorded, such as "2001", "Masayuki Abe", "80", "90", and "60", are referred to as occurrence (or instance). In addition, a set of occurrences that an attribute can take is referred to as domain. (Domain may include data types and constraint conditions of attributes.) Therefore, relation is defined as a subset of the direct product of domains. (Direct product: a set operation that obtains all combinations, Subset: a set that is included in a certain set)



Domain

Student Number
2001
2002

×

Domain

Japanese
80
60

→

Direct product

Student Number	Japanese
2001	80
2001	60
2002	80
2002	60

→

Subset

Student Number	Japanese
2001	80
2002	60

(Relation)

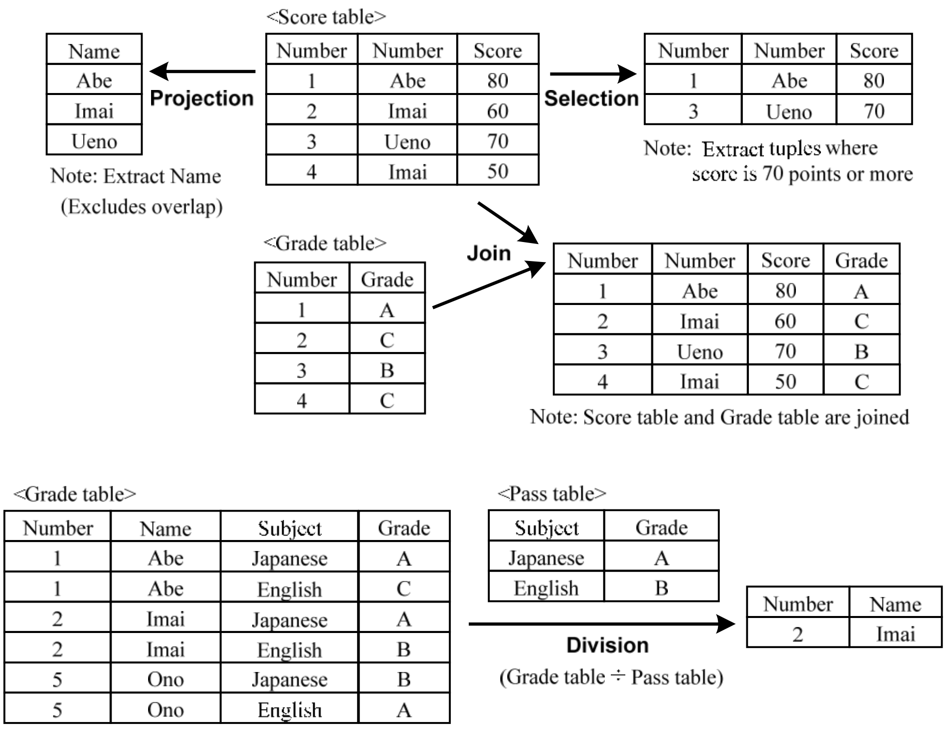
Figure 4-3 Definition of relations

The relational model has a close relationship with the set. Therefore, in the relational database that implements a relational model, the database is operated in combination with relational operations (relational algebra) that are specific to relational database and set operations that are based on the concept of set.

	Operation name	Operation contents
Relational operation	Selection	Extracts the tuples that satisfy the specified conditions.
	Projection	Extracts the specified attributes.
	Join	Combines multiple tables into one.
	Division	Extracts the tuples that are identical to all tuples in another table.
Set operation	Union	Extracts all tuples from two tables excluding any overlap.
	Difference	Extracts all tuples excluding the tuples that belong to another table.
	Product	Extracts the tuples that appear commonly in two tables.
	Direct product	Combines all tuples in two tables.

[Examples of relational operations/set operations]

- Relational operations



<Score table>

Number	Number	Score
1	Abe	80
2	Imai	60
3	Ueno	70
4	Imai	50

Name
Abe
Imai
Ueno

Note: Extract Name
(Excludes overlap)

Number	Number	Score
1	Abe	80
3	Ueno	70

Note: Extract tuples where
score is 70 points or more

<Grade table>

Number	Grade
1	A
2	C
3	B
4	C

Number	Number	Score	Grade
1	Abe	80	A
2	Imai	60	C
3	Ueno	70	B
4	Imai	50	C

Note: Score table and Grade table are joined

<Grade table>

Number	Name	Subject	Grade
1	Abe	Japanese	A
1	Abe	English	C
2	Imai	Japanese	A
2	Imai	English	B
5	Ono	Japanese	B
5	Ono	English	A

<Pass table>

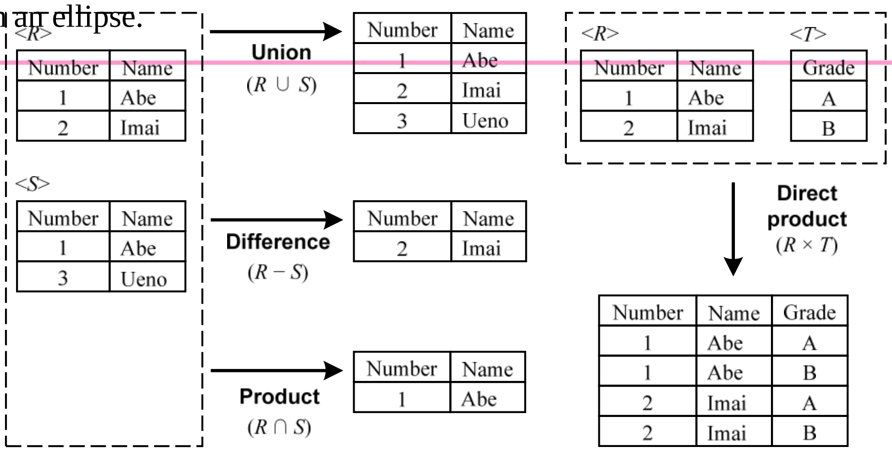
Subject	Grade
Japanese	A
English	B

Number	Name
2	Imai

(Grade table ÷ Pass table)

- Set operations

with an ellipse.



<*R*>

Number	Name
1	Abe
2	Imai

<*S*>

Number	Name
1	Abe
3	Ueno

Number	Name
1	Abe
2	Imai
3	Ueno

Number	Name
2	Imai

Number	Name
1	Abe

<*R*>

Number	Name
1	Abe
2	Imai

<*T*>

Grade
A
B

Number	Name	Grade
1	Abe	A
1	Abe	B
2	Imai	A
2	Imai	B

In addition, there are following manipulations of the relational database.

- Insert: Adding (i.e., inserting) new data (i.e., tuples) in a relation (i.e., table)
- Update: Changing (i.e., updating) the data (i.e., tuples) recorded in a relation (i.e., table)
- Delete: Deleting the data (i.e., tuples) recorded in a relation (i.e., table)

1-2-3 Conceptual Design of Databases

Conceptual design of databases is the process of creating conceptual data models. For that, it is necessary to conduct data analysis at the beginning. The data analysis identifies the data that is required for the targeted operations where database is implemented, and then the meaning and relation of this data are analyzed and summarized. In addition, data items are standardized by using rules such as naming convention in order to eliminate data duplication. Metadata such as names, meanings, or attributes of data items are recorded in the data dictionary for users in DD/D (Data Dictionary/Directory) to ensure that synonyms/homonyms do not occur. The conceptual data model is created on the basis of the results of data analysis. For the conceptual data model, it is common to use E-R model (i.e., E-R diagram) that represents the target content with two concepts of entity and relationship.

[Constituent elements of E-R model (E-R diagram)]

- Entity

  It is an object that is managed and represented with a rectangle.
- Relationship

  It is a relation between entity and entity, and it is represented with a rhombus.
- Attribute

  It is a characteristic or property of an entity and a relationship, and it is represented

In E-R model, it is possible to represent cardinality (i.e., multiplicity) of the relationship. There are three types of cardinality: “one-to-one”, “one-to-many”, and “many-to-many”. For example, Figure 4-4 in the E-R model shows a “many-to-many” relation where one instructor delivers a lecture to multiple students, and one student attends lectures of many instructors.

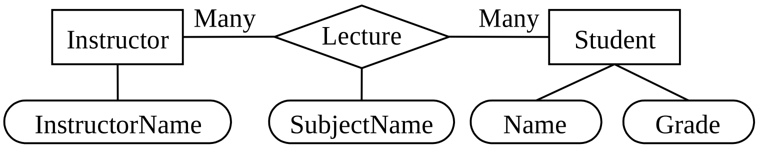


Figure 4-4 Example of E-R model (E-R diagram) (1)

When cardinality in E-R model is represented, “Many” is also represented as “*”, “M”, or “N”. Another notation method is to directly connect two entities with a line as shown in Figure 4-5. In this case, an arrow on the line means “Many.” Figure 4-5 shows that multiple types of products are purchased from a wholesaler, and products are always purchased from the same wholesaler.

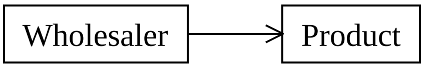


Figure 4-5 Example of E-R model (E-R diagram) (2)

1-2-4 Logical Design of Databases

Logical design of databases is the process of creating logical data models in consideration of the databases to be implemented. (Files/records/fields (items) are defined in a structured database, while tables/tuples/attributes are defined in a relational database.)

In the case of a relational database, table design is performed to decide which attributes of the table should be used for managing the data items represented in conceptual design. At that time, primary key and foreign key of each table should also be considered.

- Primary key

  It is a field (or a combination of fields) that uniquely identifies each tuple (i.e., record) in a table. (Composite key made of multiple fields is also acceptable.) For primary key, there are constraints, such as non-NULL constraint that does not allow NULL (null value) and unique constraint that does not allow duplication of values in a table.
- Foreign key

with an ellipse.

It is a field (or a combination of fields) for referencing a tuple (i.e., record) whose values match the primary key of another table. For a foreign key, it is possible to define referential constraint, which requires that the primary key of the table to be referenced must have a tuple (i.e., record) having the same value.

Note: Non-NULL constraint, unique constraint, and referential constraint are also called consistency constraint for always maintaining the data in the correct status.

In a general table design, required fields are listed in a table in the un-normalized form (UNF). Next, data normalization is performed in order to design efficient table. Functional dependency is the property where if a certain attribute is decided, other attributes is uniquely decided.

[Data normalization procedure]

1) First normalization (Deliverable: Table in the first normal form)
   Iterative fields or derived fields (i.e., fields determined through calculation) are eliminated.
2) Second normalization (Deliverable: Table in the second normal form)
   When the primary key is a composite key, fields that are dependent on some of the fields constituting the primary key are split into a separate table. (This dependency is called partial functional dependency.)
3) Third normalization (Deliverable: Table in the third normal form)
   Fields dependent on the fields other than the primary key are split into a separate table. (This dependency is called transitive functional dependency.)

By proceeding up to the third normalization as per the procedure, it is possible to design a table where all fields are dependent on the primary key. (This is called full functional dependency.) However, there are cases where the third normal form is not the best option. (For example, it may be more efficient to retain frequently used derived fields.) Therefore, it is necessary to consider including the operational aspects.

Example: Provide the third normal form obtained by normalizing the following table in un-normalized form. Underlined fields are primary keys, and { } indicates repetitive fields.

Voucher (<u>VoucherNumber</u>, Date, CustomerNumber, CustomerName, TotalAmount,
{ProductNumber, ProductName, UnitPrice, Quantity, Amount})

1) First normalization: Eliminate repetitive fields (ProductNumber,

ProductName, UnitPrice, Quantity, Amount) and derived fields (TotalAmount, Amount)

Voucher (VoucherNumber, Date, CustomerNumber, CustomerName, ProductNumber, ProductName, UnitPrice, Quantity)

2) Second normalization: Split partial functional dependency ({Date, CustomerNumber, CustomerName} dependent on VoucherNumber, {ProductName, UnitPrice} dependent on ProductNumber)

Voucher (VoucherNumber, Date, CustomerNumber, CustomerName)
Details (VoucherNumber, ProductNumber, Quantity)
Product (ProductNumber, ProductName, UnitPrice)

3) Third normalization: Split transitive functional dependency (CustomerName dependent on CustomerNumber)

Voucher (VoucherNumber, Date, CustomerNumber)
Details (VoucherNumber, ProductNumber, Quantity)
Product (ProductNumber, ProductName, UnitPrice)
Customer (CustomerNumber, CustomerName)

1-2-5 Physical Design of Databases

Physical design of databases is the process of creating physical data models in consideration of the database product (e.g., database software) to be implemented. Here, the data type that is most appropriate for each attribute is selected from the available data types in the database product, and memory format or mapping on hard disk is reviewed. In addition, required disk capacity is estimated from the anticipated data volume, and performance evaluation, such as processing efficiency or access efficiency, is conducted. At that time, it is also necessary to consider defining index (i.e., search key) for improving access efficiency. When the size of the database is large, the search efficiency can be significantly increased by defining index for the attributes that are frequently used in search operation. However, it should be noted that disk capacity for recording index is required and the process of updating the attributes of index requires more time than the usual processing of attributes.

1 - 3 DBMS (DataBase Management System)

DBMS (DataBase Management System) is software that manages databases for effectively using the databases. The person who manages the databases by using software, such as DBMS, is called the DBA (DataBase Administrator).

1-3-1 Database Definition Function

Database definition function is a function that defines schema (definition and description concerning logical structure, storage structure, and physical structure of databases). Schema is mostly defined by using the following three-schema architecture.

Schema name	Meaning	Language used
Conceptual schema (Schema)	Defines the logical structure and name of the overall database.	DDL (Data Definition Language)
External schema (Subschema)	Defines the database as seen from the user's viewpoint (only the required part of database)	DDL (Data Definition Language) DML (Data Manipulation Language)
Internal schema (Storage schema)	Defines the physical structure of databases (storage area and organization method)	DSDL (Data Storage Definition Language)

1-3-2 Database Manipulation Function

Database manipulation function is a function that offers the usage environment of database language (e.g., SQL in the case of a relational database) that is used in definition and operation of data. The following are the execution methods of database manipulation language (e.g., SQL) provided by DBMS.

- Host language system

  This is a method of executing SQL statements by using a high level language. It includes embedded SQLs that describe SQL statements in a program and module language system that creates SQL statements as external modules and calls them from the program.
- Independent language system

  This method independently executes SQL statements. It includes interactive SQL method where SQL statements are entered from the command line and command driven method where the registered SQL statements are called and executed with commands having parameters. In addition, some DBMS for PCs allow easy operation of databases where, by selecting from the predetermined forms and entering conditions, the process is executed as a query (i.e., a command to be converted into SQL statements for operating database).

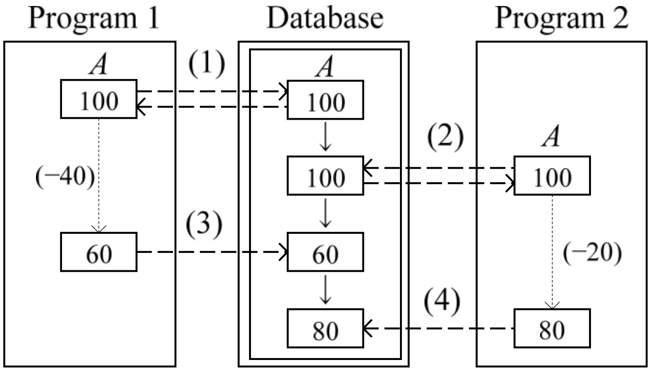
1-3-3 Database Control Function

Database control function is a function for improving reliability and security of the data recorded in the databases.

(1) Maintenance function

Maintenance function is a function for maintaining integrity of data (i.e., data integrity). The following double update is the typical example where integrity of data is lost. (As a result, inconsistency occurs.)

[Example of double update]



1) Program 1 references Data *A* (100) in the database.
2) Program 2 references Data *A* (100) in the database.
3) Data *A* is updated with the content (60) that is modified in Program 1.
4) Data *A* is updated with the content (80) that is modified in Program 2.
→ Updated content is overwritten, and update of Program 1 becomes invalid.

Exclusive control is designed for resolving such a problem. There are several methods in exclusive control, and the typical method is lock system.

The lock system locks the database record that is referenced and restricts further referencing until the process is completed. When this method is used, there is no risk of data being referenced by another program during the update, therefore, the integrity of data increases.

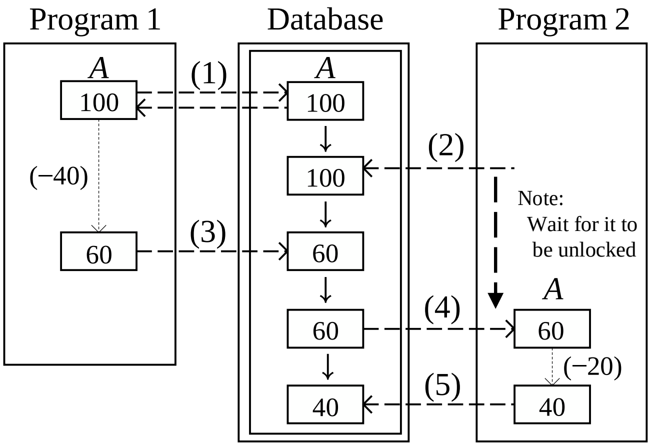
The following two types of locks are used separately in the lock system.

- Shared lock

  This lock is mainly used when data is read. In shared lock, only reading is permitted for the users other than the user who applied the lock.
- Exclusive lock

  This lock is mainly used when data is updated. Neither reading nor writing is permitted for the users other than the user who applied the lock.

[Example of exclusive control (lock system)]



1) Program 1 references Data *A* (100) in the database.
   → Apply exclusive lock to Data *A*.
2) Program 2 tries to reference Data *A* in the database. However, since Data *A* is locked, Program 2 waits for it to be unlocked.
3) Data *A* is updated with the content (60) that is modified in Program 1.
   → Release the lock of Data *A*. (Data *A* is unlocked.)
4) After the release of the lock is confirmed, Program 2 references Data *A* (60).
   → Apply exclusive lock to Data *A*.
5) Data *A* is updated with the content (40) that is modified in Program 2.
   →Release the lock of Data *A*. (Data *A* is unlocked.)

However, the problem called deadlock may occur in the lock system. Deadlock is the phenomenon where multiple programs are simultaneously in the waiting state because of the lock and their executions completely stop. It is difficult to fully avoid the deadlock. But, by reducing the unit (granularity) of applying the lock, the possibility of the occurrence of deadlock can be reduced.

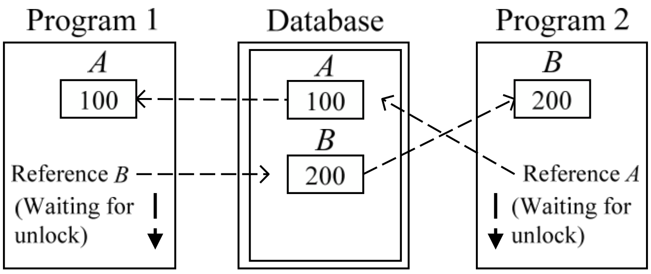


Figure 4-6　Image of deadlock

Other methods of exclusive control include semaphore system, which manages databases as resources by means of semaphore.

(2)　Security protection function

Security protection function is a function for maintaining security of data (i.e., data security). Critical data and highly confidential data are stored in databases. Therefore, data security protection becomes necessary. The following security protection functions are provided by DBMS.

- Encryption

  In this method, contents that are recorded in a database are encrypted. Even if the information of the database is leaked to a third party, that party does not know the meaning of the information. By distributing the software for decryption to only valid users, the security protection function can be increased in a relatively easy manner. However, problems, such as "decryption is not absolutely impossible" and "there is a risk of leakage of encryption/decryption software," may occur. Therefore, this method is mostly used with other methods.

- Access rights setting

  In this method, processes (i.e., accesses) are allowed for the applicable database is defined in advance. Detailed access rights can be set for each user, such as "User *A* can refer to data, but cannot update it" and "User *B* can refer to and update data, but cannot delete it." The following is the summary of various permissions with respect to database and access rights that can be set in general.

	"Read" right	"Insert" right	"Delete" right	"Update" right
Right to "connect"	Yes	Yes	Yes	Yes
Right to "search"	Yes	Yes	Yes	Yes
Right to make a "new registration"		Yes		
Right to "delete"			Yes	

Right to “update”				Yes

- Password setting

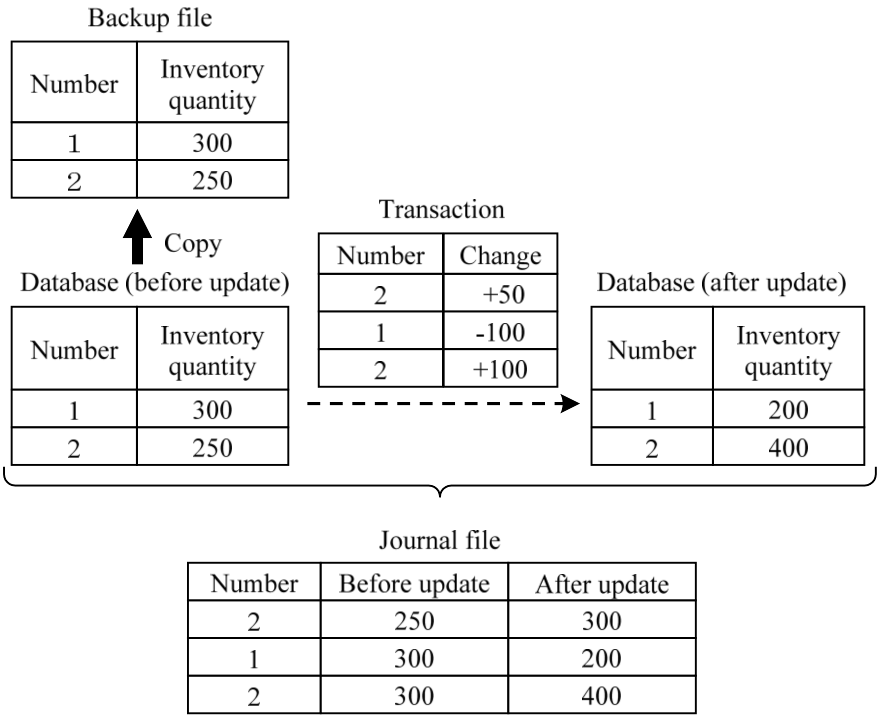
  This method uses a user ID and a password to check whether the user of the database has the right to use the database or not. It is useful for preventing unauthorized use from outside. Even when the access right is set, it is mandatory to set a password in order to prevent the unauthorized use of user ID.

- Registration in journal file (log file)

  This method records the status of use (date, account, details of use) of the database and checks whether there has been any unauthorized use or not. While the journal file does not offer immediate security protection, it is useful because unauthorized users can be identified at a later date. It is mostly used in a supplementary sense.

(3) Failure recovery function

Failure recovery function is a function that restores databases at the occurrence of a failure. Failure recovery uses files, such as backup file where a database at a particular time is copied as it is, and journal file (i.e., log file) where an update process performed on the database is recorded. Journal records are created in the memory and are written in a file at the timing of commit (i.e., process of finalizing the update process) which is performed when the database update process for each transaction is completed. A log is prepared by setting a checkpoint for a certain time interval (or a certain processing amount). The database being updated in the memory is recorded in the checkpoint file. With this, it is possible to recover from the checkpoint time when a failure has occurred.



Backup file

Number	Inventory quantity
1	300
2	250

Database (before update)

Number	Inventory quantity
1	300
2	250

Transaction

Number	Change
2	+50
1	-100
2	+100

Database (after update)

Number	Inventory quantity
1	200
2	400

Journal file

Number	Before update	After update
2	250	300
1	300	200
2	300	400

Figure 4-7　Backup file and journal file

The following are two types of recovery processes performed for failure recovery.

- Rollforward

  This recovery technique is mainly used for physical failure of storage media. Contents of the backup file (or the checkpoint file) are updated on the basis of information (also known as “redo” information) after the update of the journal file, in order to restore the contents as the database at the time the failure occurred.

- Rollback

  This recovery technique is mainly used for logical faults, such as transaction error. The database just after the disabling process is restored to the status before the disabling process on the basis of the information (also known as “undo” information) before the update of the journal file. Rollback is also used in the sense of canceling the update process that has not been committed yet.

The recovery process is also performed for each transaction. In this case, transactions committed after a checkpoint are recovered to the status after completion of commit with rollforward. On the other hand, for the transactions that are running at the time the failure occurred, processes that are not committed are canceled (i.e., rolled back) and transactions are executed again after the database is reverted to the status before the process started.

In the case of Figure 4-8, T1 and T2 committed after the checkpoint are recovered with rollforward from the checkpoint time. Meanwhile, T3, which was being updated at the time the failure occurred, is executed again when the database is reverted to the status before the process started with rollback from the checkpoint time.

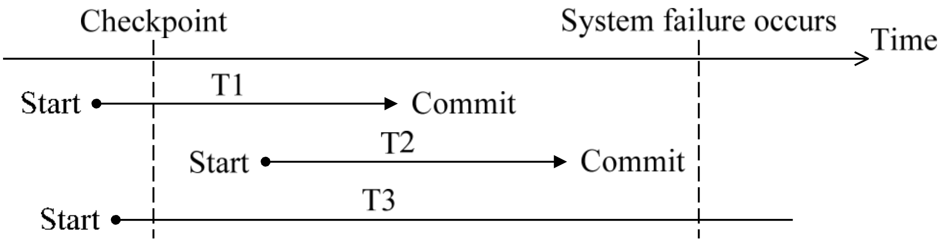


Figure 4-8 Failure recovery for each transaction

The following two methods are available for restarting the database.

- Warm start method

  This method restarts the database, while the status of memory is maintained, by using software reset where the power supply is not turned off. Without initializing the database, it is restored to the status just before it was restarted by using the checkpoint file or journal file.

- Cold start method

  This method restarts the database, after the status of memory is completely cleared, by using hardware reset where the power supply is turned off. After the database is restarted, it is restored to the status just before it was restarted by using the latest backup file and journal file.

A repetitive update (i.e., addition or deletion) of the database may result in a wasteful storage area that is not reused, which may decrease the access efficiency. In such a case, reorganization that optimizes the database is also one of the failure recovery functions.

[Transaction management of database]

Transaction processing with respect to the database is required to have ACID characteristics.

- Atomicity: The process completes by commit or rollback.
- Consistency: Contradiction (i.e., inconsistency of data) does not occur before and after the process.
- Isolation: Multiple transactions do not mutually interfere.
- Durability: Results after commit are continuously retained even after the process.

2 SQL

SQL (Structured Query Language) is a database language for using a relational database. SQL consists of DDL (Data Definition Language) that defines data, DML (Data Manipulation Language) that operates data, and others. The basic method of using SQL is explained with the following tables.

<Student>

StudentNumber	Name	Gender
6724	Kazuki Yamamoto	Male
6725	Jyugo Motoyama	Male
6816	Mone Yamada	Female
6817	Chiyo Yamamoto	Female
Alphanumeric characters	Alphabetical characters	Alphabetical characters

<Subject>

SubjectNumber	SubjectName
K11	English I
K12	English II
K21	Mathematics
Alphanumeric characters	Alphabetical characters

<Score>

StudentNumber	SubjectNumber	Score	ExaminationDate
6724	K11	65	20XX-10-20
6724	K21	85	20XX-10-21
6725	K21	60	20XX-10-21
6817	K11	85	20XX-10-20
6817	K12	90	20XX-10-20
6817	K21	95	20XX-10-21
Alphanumeric characters	Alphanumeric characters	Integer	Date

Note:
Fields underlined in each table show the primary key.

2 - 1 Data Definition

Data definition refers to defining the database, table (i.e., base table), view (i.e., virtual table), and so on.

2-1-1 Definition of Database

CREATE DATABASE statement is used for defining database.

```
CREATE DATABASE name
```

ex01: Define the database “PerformanceManagementDB”.

```
CREATE DATABASE PerformanceManagementDB
```

2-1-2 Definition of Tables

CREATE TABLE statement is used for defining a table.

```
CREATE TABLE name
```
(Column name 1, Data type 1, Column name 2, Data type 2, ...)

- The following are the main data types that can be defined.

Data type	Definition name	Meaning
Character type	CHAR	Character of the specified length (1 byte)
	NCHAR	Character of the specified length (2 bytes)
Numeric type	INT	Integer having precision defined by the processing system.
	DEC	Numerical values having integer part and decimal part of the specified number of digits.
Date type	DATE	Year-Month-Date

- `PRIMARY KEY` : Declaration of primary key (Primary key constraint: non-NULL constraint + unique constraint)
- `FOREIGN KEY` : Declaration of foreign key (referential constraint)
- `NOT NULL` : non-NULL constraint (NULL is not allowed as occurrence)
- `UNIQUE` : unique constraint (Duplication of occurrence in the table is not allowed))
- `CHECK` : check constraint (specifies the conditions of occurrence)

Note: It is called column constraint when it is simultaneously performed with column definition and table constraint when it is performed at the end.

ex02: Define table “Score”.

```
CREATE TABLE Score
 ( StudentNumber    CHAR(4),
   SubjectNumber    CHAR(3),
   Score            INT  CHECK (Score >= 0),
   ExaminationDate  DATE NOT NULL,
   PRIMARY KEY (StudentNumber, SubjectNumber),
   FOREIGN KEY (StudentNumber) REFERENCES Student(StudentNumber),
   FOREIGN KEY (SubjectNumber) REFERENCES Subject(SubjectNumber)
  )
```

Notes:

* A check constraint (as column constraint) is set for the score to be 0 and above.

* Non-NULL constraint (as column constraint) is set for ExaminationDate.

* Primary key constraint (as table constraint) is set for (StudentNumber, SubjectNumber).

* For StudentNumber, a referential constraint (as table constraint) is set so that StudentNumber in Student is referenced as a foreign key.

* For SubjectNumber, a referential constraint (as table constraint) is set so that SubjectNumber in Subject is referenced as a foreign key.

Moreover, use the following ALTER TABLE statement when the configuration of the defined tables (i.e., database) is changed (or redefined).

```
ALTER TABLE name Details of redefinition
```

Note: In the underlined part, write `ADD` when attribute is added, write `MODIFY` when data type is changed, and so on.

2-1-3 Definition of View

View refers to the table that is virtually set from the real tables and corresponds to external schema. (View is also referred to as virtual table, and table is also referred to as base table.) View can record attributes of a single table and new attributes that are created from multiple tables. Users can operate the defined views in the same way as normal tables, and therefore, restrict the scope of use of a database, which can help in data security protection and maintenance.

CREATE VIEW statement is used for defining a view.

```
CREATE VIEW Name AS SELECT …
```

Note: In the underlined part, write the SELECT statement for extracting data. (Details are explained in 2-2.)

ex03: Extract StudentNumber and Name from the “Student” table and define the view “Name”.

```
CREATE VIEW Name AS SELECT StudentNumber, Name FROM Student
```

Note: Underlined part is the SELECT statement that extracts StudentNumber and Name from the Student table.

View defines only a virtual table and does not create any new tables. (It can simply be considered that the shaded portion in Figure 4-9 is made invisible.) Therefore, updating a view would also reflect the updated results in the original table. Because of that, there is a constraint that the views where tuples and the original table are not in a 1:1 relationship cannot be updated. Such a view (e.g. view that uses the set function) is described later.

StudentNumber	Name	Gender
6724	Kazuki Yamamoto	Male
6725	Jyugo Motoyama	Male
6816	Mone Yamada	Female
6817	Chiyo Yamamoto	Female

Figure 4-9　Concept of view

2-1-4 Definition of Access Right

Access right is the right (i.e., processing privilege) for each user to use a database. Setting access right is useful for data security protection.
GRANT statement is used for defining access right.

```
GRANT Privilege 1, Privilege 2, ••• ON Table Name TO Identifier
```

- The following access rights can be defined.

Type of Privilege	Meaning
SELECT	Permission to refer to the database
INSERT	Permission to add (i.e., insert) data in the database
DELETE	Permission to delete data from the database
UPDATE	Permission to update data in the database
REFERENCES	Permission to redefine the database

ALL PRIVILEGES	All permission related to the database

Note: Identifier uniquely specifies the intended person who is granted the permission.

ex04: Set the permission to refer and update the "Student" table in the identifier "EducationAffairsOffice".

```
GRANT SELECT, UPDATE ON Student TO EducationAffairsOffice
```

REVOKE statement is used for canceling the defined (i.e., granted) permission.

```
REVOKE Privilege 1, Privilege 2, ••• ON Table name TO Identifier
```

2-1-5 Data Storage

Data storage is the process of inserting data in a table. (While it should be included in data manipulation, it is explained in data definition as preparation for using the database.) There are two methods of storing data in a table. Generally, method 1) is suitable for small amounts of data while method 2) is more efficient for large amounts of data.

1) Storing data in units of tuples (i.e., records)
   INSERT statement is used for storing data in units of tuples (i.e., records).

```
INSERT INTO Table name VALUES (Data 1, Data 2, •••)
```

ex05: Store tuple (K11, English I) in the "Subject" table.

```
INSERT INTO Subject VALUES ('K11', 'English I')
```

2) Using data storage program
   A data storage program is prepared in the host language system beforehand, or the data storage program that is available in the utility program is used.

2 - 2 Data Manipulation

Data reference (i.e., extraction) from the database is the most widely-used data manipulation. In SQL, use the SELECT statement for referring to data. The explanation provided here is mainly focused on the method of using the SELECT statement.

2-2-1 Reference Without Specifying Conditions

Reference without specifying conditions corresponds to the relational operation “Projection” that extracts the specified attributes. Use the following SELECT statement in reference without specifying conditions.

```
SELECT Column name 1, Column name 2, ••• FROM Table name
```

Notes:

1. If “*” is specified as column name, all columns are extracted.
2. When `DISTINCT` is specified in a column name, a duplication is eliminated. (The same results are not be displayed.)

ex06: Extract all columns from the “Student” table.

```
SELECT * FROM Student
```

<Execution results of ex06>

StudentNumber	Name	Gender
6724	Kazuki Yamamoto	Male
6725	Jyugo Motoyama	Male
6816	Mone Yamada	Female
6817	Chiyo Yamamoto	Female

ex07: Extract Gender from the “Student” table. (Projection).

```
SELECT Gender FROM Student
```

ex08: Extract Gender from the “Student” table after duplication is eliminated. (Projection)

```
SELECT DISTINCT Gender FROM Student
```

<Execution results of ex07>

Gender
Male
Male
Female
Female

<Execution results of ex08>

Gender
Male
Female

2-2-2 Reference With Specifying Conditions

Reference with specifying conditions corresponds to the relational operation “Selection” that extracts the tuples that satisfy the specified conditions. Use the following SELECT statement in reference with specifying conditions.

```
SELECT Column name 1, Column name 2, ••• FROM Table name WHERE Extraction conditions
```

Notes: 1. Tuples that satisfy the conditions that are described in the WHERE clause are extracted.

2. The following are the main operators and notations used in extraction conditions.

<Comparison operators>

Notation	Example of use	Meaning
=	$A = B$	*A* is equal to *B*.
<>	$A <> B$	*A* is not equal to *B*. ($A \neq B$)
<	$A < B$	*A* is smaller than *B*.
<=	$A <= B$	*A* is equal to or smaller than *B*. ($A \leq B$)
>	$A > B$	*A* is greater than *B*.
>=	$A >= B$	*A* is equal to or greater than *B*. ($A \geq B$)

<Logical operators>

Notation	Meaning
AND	True if both the conditions are satisfied
OR	True if either of the conditions is satisfied
NOT	False if the original condition is true, and true if it is false (negation)

<Others (specifying special conditions)>

Notation	Meaning
BETWEEN *A* AND *B*	Specifies the range of values (*A* or more, and *B* or less)
LIKE …	Specifies the pattern of string (wild card specification) (%: Characters above 0 characters, _: 1 character)
IS NULL	Judgment of NULL

ex09: From the “Student” table, extract tuples where Gender is 'Female' (Selection).

```
SELECT * FROM Student WHERE Gender='Female'
```

ex10: From the “Subject” table, extract subject names other than the SubjectName 'Mathematics' (Selection, Projection).

```
SELECT SubjectName FROM Subject WHERE SubjectName<>'Mathematics'
```

<Execution results of ex09>

StudentNumber	Name	Gender
6816	Mone Yamada	Female
6817	Chiyo Yamamoto	Female

<Execution results of ex10>

SubjectName
English I
English II

ex11: From the “Score” table, extract StudentNumber, SubjectNumber, and Score where SubjectNumber contains '2' and Score is between 80 and 90.

```
SELECT StudentNumber, SubjectNumber, Score FROM Score
  WHERE (SubjectNumber LIKE '%2%')
                              AND (Score BETWEEN 80 AND 90)
```

ex12: From the “Score” table, extract StudentNumber, SubjectNumber, and Score where SubjectNumber contains '2' or Score is between 80 and 90.

```
SELECT StudentNumber, SubjectNumber, Score FROM Score
  WHERE (SubjectNumber LIKE '%2%')
                              OR (Score BETWEEN 80 AND 90)
```

<Execution results of ex11>

Student Number	Subject Number	Score
6724	K21	85
6817	K12	90

<Execution results of ex12>

Student Number	Subject Number	Score
6724	K21	85
6725	K21	60
6817	K11	85
6817	K12	90
6817	K21	95

ex13: From the “Score” table, among the tuples where the score is greater than 60, extract the tuples where the 2nd digit as counted from the left edge of StudentNumber is '7' or ExaminationDate is '20XX-10-21'.

```
SELECT * FROM Score
  WHERE Score > 60
     AND (StudentNumber LIKE '_7_ _' OR ExaminationDate = '20XX-10-21')
```

<Execution results of ex13>

StudentNumber	SubjectNumber	Score	ExaminationDate
6724	K11	65	20XX-10-20
6724	K21	85	20XX-10-21
6817	K21	95	20XX-10-21

Note: If ( ) is not included, AND is evaluated first, and incorrect results are extracted as shown below.

StudentNumber	SubjectNumber	Score	ExaminationDate
6724	K11	65	20XX-10-20
6724	K21	85	20XX-10-21
6725	K21	60	20XX-10-21
6817	K21	95	20XX-10-21

2-2-3 Grouping of Data

Grouping of data refers to handling the tuples where a certain attribute has the same value in a consolidated manner, and functions used for this are referred to as set functions (or aggregate functions).

GROUP BY clause is used for grouping of data.

```
SELECT Extraction items FROM Table name GROUP BY Grouping column name
```

- In extraction items, grouping column names, set functions, and constants can be described.
- The following are the typical set functions.

Set function	Meaning
SUM	Determine the sum of group.
AVG	Determine the average of group.
MIN	Determine the minimum value of group.
MAX	Determine the maximum value of group.

COUNT	Count the number of records (i.e., number of tuples) in the group. * ··· Count all tuples. DISTINCT Column name ··· Count the tuples where the value of the column does not overlap.

- **AS** : A new column name can be specified in the column determined by using set functions.
- **HAVING** : The extraction condition that uses set functions can be written.

ex14: From the “Score” table, determine and extract average score of each subject.

```
SELECT SubjectNumber, AVG(Score) FROM Score
  GROUP BY SubjectNumber
```

ex15: From the “Score” table, determine and extract the number of subjects for which the respective student took an examination.

```
SELECT StudentNumber, COUNT(*) AS NumberOfSubjects FROM Score
  GROUP BY StudentNumber
```

ex16: From the “Score” table, determine and extract the number of days for which the respective student took an examination.

```
SELECT StudentNumber, COUNT(DISTINCT ExaminationDate) AS NumberOfDays
    FROM Score  GROUP BY StudentNumber
```

<Execution results of ex14>

SubjectNumber	AVG (Score)
K11	75
K12	90
K21	80

<Execution results of ex15>

StudentNumber	NumberOfSubjects
6724	2
6725	1
6817	3

<Execution results of ex16>

StudentNumber	NumberOfDays
6724	2
6725	1
6817	2

ex17: From the "Score" table, extract students having TotalScore (total of score) of 150 or more.

```
SELECT StudentNumber, SUM(Score) AS TotalScore FROM Score
  GROUP BY StudentNumber HAVING SUM(Score) >= 150
```

<Execution results of ex17>

StudentNumber	TotalScore
6724	150
6817	270

2-2-4 Sorting of Data

Sorting of data refers to rearranging the extraction results in the specified order of a particular attribute. (When there are no sorting instructions, data is extracted in the order in which it is recorded in the original table.)
ORDER BY clause is used for sorting data.

```
SELECT Column name 1, Column name 2, ••• FROM Table name
  ORDER BY Column name to be sorted    Sorting order
```

- The following two types of sorting orders can be specified.

Sorting order	Meaning
ASC	Sort in the ascending order. (This is the default value when sorting order is omitted.)
DESC	Sort in the descending order.

- For records having the same column value, original recording order is generally retained.
- The column name to be sorted can also be specified by indicating how many columns from the left it is.

ex18: Sort and extract the "Score" table in the descending order of score.

```
SELECT * FROM Score ORDER BY Score DESC
```

ex19: From the "Score" table, determine the highest score for each student and extract in the ascending order.

```
SELECT StudentNumber, MAX(Score) FROM Score
  GROUP BY StudentNumber  ORDER BY 2
```

＜Execution results of ex18＞

Student Number	Subject Number	Score	ExaminationDate
6817	K21	95	20XX-10-21
6817	K12	90	20XX-10-20
6724	K21	85	20XX-10-21
6817	K11	85	20XX-10-20
6724	K11	65	20XX-10-20
6725	K21	60	20XX-10-21

＜Execution results of ex19＞

Student Number	MAX(Score)
6725	60
6724	85
6817	95

2-2-5 Joining the Tables

Joining the tables means combining two or more tables into one table. There are a few methods of joining. However, the easiest method is explained by using the following two tables.

＜α＞

X	Y
X1	Y2
X2	Y1

＜β＞

Y	Z
Y1	Z1
Y2	Z2

When two or more tables are used, list all tables used in the FROM clause. With this, the combinations (i.e., direct product) of all tuples of the specified tables are created. This method of joining, which takes the direct product of two or more tables, is called cross join.

```
[Cross join] SELECT * FROM α, β
```

When two or more tables are joined, values of the columns having common meaning mostly combine the same tuples. Therefore, with the condition of the WHERE clause, extract the tuples having the same values of column from the results of cross join. (Joining the tuples having the same value of the corresponding column is called equijoin.)

```
[Equijoin]  SELECT * FROM α, β WHERE α.Y = β.Y
```

<Execution results of cross join>

α.X	α.Y	β.Y	β.Z
X1	Y2	Y1	Z1
X1	Y2	Y2	Z2
X2	Y1	Y1	Z1
X2	Y1	Y2	Z2

<Execution results of equijoin>

α.X	α.Y	β.Y	β.Z
X1	Y2	Y2	Z2
X2	Y1	Y1	Z1

ex20: From the "Score" table and the "Subject" table, extract StudentNumber, SubjectName for which the students took an examination, and Score.

```
SELECT StudentNumber, SubjectName, Score FROM Score, Subject
  WHERE Score.SubjectNumber = Subject.SubjectNumber
```

<Execution results of ex20>

Student Number	SubjectName	Score
6724	English I	65
6724	Mathematics	85
6725	Mathematics	60
6817	English I	85
6817	English II	90
6817	Mathematics	95

Note:
When columns that are common in two tables are extracted, it is necessary to clearly identify the table to which the rows belong.
Example: `SELECT Score. SubjectNumber,` …

When the table with FROM clause is specified, by adding correlation name, SQL statement in ex20 can be simplified as follows:

```
SELECT StudentNumber, SubjectName, Score FROM Score X, Subject Y
  WHERE X.SubjectNumber = Y.SubjectNumber
```

One of the methods of joining tables uses JOIN specification in the FROM clause. With JOIN specification, the following three joining operations can be achieved.

- `CROSS JOIN` (Cross join)
  It specifies direct product of multiple tables. (The result is the same as listing the tables.)
- `INNER JOIN` (Inner join)
  It specifies equijoin of tables with joining conditions.

```
SELECT StudentNumber, SubjectName, Score
  FROM Score X INNER JOIN Subject Y
```

```
                              ON X.SubjectNumber = Y.SubjectNumber
```

- OUTER JOIN (Outer join)

  It extracts all tuples of the priority table even if the joining conditions are not met.

  LEFT OUTER JOIN : Table on the left is the priority table.

  RIGHT OUTER JOIN: Table on the right is the priority table.

2-2-6 Sub Reference (Subquery)

Sub reference (subquery) refers to searching for another table on the basis of the search results of a certain table. SELECT statement that determines the conditions is called subquery statement, while the SELECT statement that searches for the table by using the results of subquery statement is called main query statement. Subquery is broadly classified into two types.

(1) Subquery statement that can be independently executed

This subquery statement is independently executed for getting the conditions of the main query statement. In this format, subquery statement inside the parentheses ( ) is executed first, and main query statement is executed on the basis of its results.

```
SELECT Column name, ••• FROM Table name
  WHERE Column name  Comparison term  (subquery statement)
```

- Comparison term includes IN predicate, and ANY predicate and ALL predicate combined with comparison operator.

Predicate	Meaning
IN	Either of the results of subquery statement matches. (Same as “= ANY”)
ANY	Comparison operator holds true for either of the results of subquery statement.
ALL	Comparison operator holds true for all results of subquery statement.

Note: If the result of subquery statement is always one, a single comparison operation can also be used.

ex21: From the “Student” table and the “Score” table, extract the names where ExaminationDate is '20XX-10-20'.

```
SELECT Name FROM Student
  WHERE StudentNumber IN ( SELECT StudentNumber FROM Score
                            WHERE ExaminationDate = '20XX-10-20' )
```

<Execution results of ex21>

Name
Kazuki Yamamoto
Chiyo Yamamoto

Note: Results of the subquery statement have the same meaning as the following SQL statement.

```
SELECT Name FROM Student
  WHERE  StudentNumber  IN  ('6724',  '6817',
  '6817')
```

(2) Correlation subquery (subquery statement that cannot be executed independently)

Correlation subquery uses items of the table in the main query statement as a condition, and therefore, this subquery statement cannot be executed independently. In this case, tuples are fetched one by one from the table in the main query statement and the subquery statement is executed. On the basis of whether the result of the subquery statement is one row or more, whether to extract the concerned tuple or not is decided. EXISTS predicate is used in this decision.

ex22: From the "Student" table and the "Score" table, extract the names where ExaminationDate is '20XX-10-20'.

```
SELECT Name FROM Student X
  WHERE EXISTS ( SELECT * FROM Score Y
     WHERE X.StudentNumber = Y.StudentNumber
                     AND ExaminationDate = '20XX-10-20' )
```

In the SQL statement of ex22, the subquery statement inside the parentheses ( ) cannot be executed independently. (Correlation name X cannot be interpreted.) Therefore, tuples of the "Student" table are fetched one at a time, and the subquery statement is executed.

- Execute the subquery statement for the tuple of StudentNumber '6724'.

```
SELECT * FROM Score Y
   WHERE '6724' = Y.StudentNumber AND ExaminationDate = '20XX-10-20'
```

<Execution results>

StudentNumber	SubjectNumber	Score	ExaminationDate
6724	K11	65	20XX-10-20

=> The result is one row or more, and therefore, the tuple of StudentNumber '6724' is extracted.

- Execute the subquery statement for the tuple of StudentNumber '6725'.

```
SELECT * FROM Score Y
  WHERE '6725' = Y.StudentNumber AND ExaminationDate = '20XX-10-20'
```

<Execution results>

StudentNumber	SubjectNumber	Score	ExaminationDate

=> Since there is no single row in the result, the tuple of StudentNumber '6725' is not extracted.

In this manner, by executing the subquery statement by using StudentNumber of all tuples, tuples to be extracted are decided.

When this example is represented without using subquery, the SQL statement is described as below. Here, note that the DISTINCT specification is omitted and (Chiyo Yamamoto) is extracted twice.

```
SELECT DISTINCT Name FROM Student X, Score Y
  WHERE X.StudentNumber = Y.StudentNumber
                AND ExaminationDate = '20XX-10-20'
```

In addition, IN predicate and EXISTS predicate can be used in combination with the logical operator NOT.

- `NOT IN:`   True if only unequal subquery rows are found
- `NOT EXISTS:` True if tuple does not exist as the result of subquery

2-2-7 Other Methods of Using SQL

(1) Other data manipulation languages

- INSERT statement: This statement is used when a tuple is inserted (i.e., added) into the table.

```
INSERT INTO Table name VALUES ( Data 1, Data 2, ••• )
```

- UPDATE statement: This statement is used when the data in the table is updated (i.e., changed).

```
UPDATE Table name SET Column name = Updated value
       WHERE Update specification condition
```

- DELETE statement: This statement is used when a tuple is deleted from the table.

```
DELETE FROM Table name WHERE Delete specification condition
```

(2) Cursor

Cursor is used when data in the database is used in the host language system (i.e., embedded SQL). Usually, host language (i.e., programming language) handles data in units of records, while SQL handles data in units of sets. Therefore, the cursor acts as a bridge by fetching and transferring one row at a time.

- DECLARE CURSOR statement: This statement defines the derived table to be processed and assigns the cursor.

```
DECLARE Cursor name CURSOR FOR SELECT …
```

Note: Dynamic SQL is also available that prepares the SELECT statement at the time of execution by using the PREPARE statement.

- OPEN statement: This statement starts the process of cursor.

```
OPEN Cursor name
```

- FETCH statement: This statement reads data from the position of cursor.

```
FETCH Cursor name INTO Variable name
```

- CLOSE statement: This statement terminates the process of cursor.

```
CLOSE Cursor name
```

3 Various Databases

In present day society, database technology is applied in various areas. In companies, databases are used in many systems, such as corporate accounting systems, inventory control systems, document management systems, and sales support systems. In this section, we will learn about the database technology used in these systems.

3 - 1 Distributed Database

Distributed database is a technology that handles databases distributed at multiple sites as if they were one database. Databases are, by nature, used for the purpose of consolidating data and managing it in an integrated manner. However, there are following problems concerning databases consolidated over a network: the fault of a database affects the entire system, the network becomes over crowded because of concentration of processes, and maintenance/management of the database becomes difficult. Therefore, the concept of distributed database was developed where databases are dispersed and constructed at multiple sites.

In the distributed database, by using the RDA (Remote Database Access) system, users can use the database without becoming aware of which database they are accessing. (This is called transparency of distributed database.) However, just like usual distributed processing systems, it suffers from the problem that operations, such as security management, take a lot of effort. In addition, it is possible that data may duplicated. Therefore, there is a risk that consistency of data is lost. In the distributed database, these problems are resolved with the following techniques.

- Two-phase commitment (Two-phase commit)

  This is a commitment control method that executes the update process in the distributed database after the process is divided into two phases. It is suitable when an instantaneous update is required. The control method that instructs COMMIT or ROLLBACK without checking whether the update process can be performed or not is called one-phase commitment.

  1) 1st phase

     Check the possibility of update process for all databases.

  2) 2nd phase

     Finalize (commit) if all databases can be updated (i.e., if all databases respond with a positive responses), and cancel (i.e., rollback) if even only one database cannot be updated (i.e., if even only one database responds with a negative response.).

- Replication

  This method places a copy (i.e., replica) of the original database as the distributed database and reflects the update done in the original database into the replicated database at regular intervals (or specified times). It is suitable when an instantaneous update is not required.

3 - 2 Data Warehouse

Data warehouse is a company-wide integrated information system (multi-dimensional database) that expands the functions of databases and extracts the required decision-making information for the business strategy of the company. It is used in OLAP (OnLine Analytical Processing), which extracts the data that is accumulated through OLTP (OnLine Transaction Processing) in the mission critical systems, builds a massive database for information analysis, and conducts multifactor analysis and forecasts of data,.

- ETL (Extract/Transform/Load)

  It refers to extracting the raw data stored in the mission critical system, processing the raw data into an easy-to-use format with tasks such as name identification and standardization, and exporting it to a data warehouse. It is also software that supports this series of processes.
- Data cleansing (Data cleaning)

  It refers to removing duplication and inconsistency of notations from the database. It is used in database optimization and ETL processing.

Data mining is the method to analyze data or regularity required for the management by using mathematical and statistical techniques such as OLAP. Mining, which is literally described as digging up data from databases, is useful in marketing strategies by analyzing a large amount of accumulated data and finding the potential needs of customers. For example, from a large amount of sales data, it extracts rules, such as “People who buy bread also buy milk simultaneously.”

- Star schema

  It is a schema of a database for analysis. It places analysis values in a radial manner focusing on the target of analysis. Creating index for implementing star schema is one of the preparations required for data mining.
- Cluster analysis

  It is an analytical technique that the similarity of data to be analyzed is quantitatively (e.g., by using distance or extent of similarity) determined in order to group data. Dendrogram is used for showing the results of analysis.

This series of data management/analysis work from building a data warehouse to data mining is also called data administration.

In recent years, the amount of accumulated data has become so large that it is also referred to as big data. It has become difficult to handle so much data with the commercially available database software (i.e., DBMS). Therefore, in some cases, a small-scale data warehouse (i.e., data mart) is used, which is prepared by extracting only data required for a specific department (or business operations) from the data warehouse. Data mart is easy to use because data is refined as per the requirement and level of end user. However, there is a risk that it may overlook unforeseen regularity.

3 - 3 Other Related Techniques

(1) IRDS (Information Resource Dictionary System)

IRDS (Information Resource Dictionary System) is the system that registers and manages metadata, such as attribute, meaning, and storage location of data, in DD/D (Data Dictionary). It is also used as a repository that manages source code or such other data, in an integrated manner in software development and maintenance.

(2) Special databases

- Commercial database

  It is a database that contains independently collected information and that is offered to third parties on a chargeable basis for generating profits. In addition, in some commercial databases, by using a thesaurus (i.e., a dictionary where synonyms and equivalent words are systematically classified and arranged), thesaurus search can search with minimum oversight with respect to various representations.

- OODB (Object-Oriented DataBase)

  It is a database that is managed with objects where data and processing (i.e., procedures) are integrated (i.e., encapsulated). Users can process the data by simply giving specific instructions (i.e., messages).

- Multimedia database

  It is a database that can handle information (e.g., images, audio) in addition to characters and numeric values. It is very hard for users to use the database by being aware of various pieces of information such as images and audio, and therefore, the concept of OODB (Object-Oriented DataBase) is used in most of the cases.

- Hypertext database

It is a network structured database that manages hypertext (e.g., a document that can freely search/display the relevant information by using keywords) used in the Internet. These days, a hypermedia database that can handle images and audio is also available.

- XML database

It is a database that can manage the document structure (e.g., tags) of XML as it is. This database uses the features of XML and has excellent flexibility and expandability.

(3) Database integration techniques

- Data mapping

It refers to creating a table (i.e., data map) that correlates data between different applications. It is used for correlating a database of different DBMSs.

- JDBC (Java DataBase Connectivity)

This API is used for accessing databases from Java. It allows the development of a highly general-purpose program that does not depend on DBMS.

Chapter 4 Exercises

Q1

Which of the following is a benefit that can be expected by installing a database system?

a) Mitigation of code design activity b) Reduction of duplicate data
c) High-speed data transfer d) Dynamic access to data

Q2

Which of the following is an appropriate explanation of the relational database?

a) Data is viewed as a two-dimensional table to users. Relation between records is associated by using the values of items in each record.
b) Relation between records is represented by using a one-to-many relationship of parent and child.
c) Relation between records is represented with the network structure.
d) A user recognizes and represents the logical layout of records.

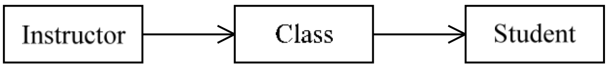
Q3

Which of the following is an appropriate explanation of “projection” that is one of relational operations?

a) From the table, it selects tuples that satisfy the given conditions and creates a new table.
b) From the table, it extracts only the specified attributes and creates a new table.
c) It extracts tuples that commonly exist in two tables and creates a new table.
d) From tuples in two tables, it joins the pairs of tuples that meet the conditions and creates a new table.

Q4

Which of the following is the appropriate interpretation of the E-R model shown below?



a) One student is looked after by one instructor.
b) One student belongs to multiple classes.
c) One instructor looks after only one class.
d) One class is looked after by two instructors, namely the homeroom teacher and deputy homeroom teacher.

Q5

Which of the following is the second normal form of the "SkillRecord" table? Here, the underlined parts show primary key, while { } shows iterative items.

SkillRecord
(EmployeeNumber, EmployeeName, {SkillCode, SkillName, SkillExperienceYears })

a)

$\underline{\text{EmployeeNumber}}$	EmployeeName	$\underline{\text{SkillCode}}$	SkillName	SkillExperienceYears

b)

$\underline{\text{EmployeeNumber}}$	EmployeeName	$\underline{\text{SkillCode}}$	SkillExperienceYears

c)

$\underline{\text{SkillCode}}$	SkillName

$\underline{\text{EmployeeNumber}}$	$\underline{\text{SkillCode}}$	SkillExperienceYears

$\underline{\text{EmployeeNumber}}$	EmployeeName

d)

$\underline{\text{SkillCode}}$	SkillName

$\underline{\text{EmployeeNumber}}$	$\underline{\text{SkillCode}}$

$\underline{\text{EmployeeNumber}}$	EmployeeName	SkillExperienceYears

Q6

Which of the following is an appropriate description of the exclusive control function of a DBMS?

a) It should be used during online updates, and needs not be used during updates with batch processing.
b) It can only be performed for each relation (i.e., table).
c) It prevents the loss of data integrity because of a lost update or such other problem.
d) The purpose of the lock method which is one of the exclusive controls is to prevent deadlock.

Q7

When a failure occurs in database operation, which of the following is the failure recovery process that returns the status of database to the status before the start of the transaction?

a) Reorganization
b) Checkpoint
c) Rollback
d) Rollforward

Q8

Which of the following is the SQL statement that returns the largest value when it runs on the 'DeliveryRecord' table?

DeliveryRecord

ProductNumber	Date	Quantity
NP200	20XX-10-10	3
FP233	20XX-10-10	2
NP200	20XX-10-11	1
FP233	20XX-10-11	2

a) `SELECT AVG(Quantity) FROM DeliveryRecord WHERE ProductNumber = 'NP200'`
b) `SELECT COUNT(*) FROM DeliveryRecord`
c) `SELECT MAX(Quantity) FROM DeliveryRecord`
d) `SELECT SUM(Quantity) FROM DeliveryRecord WHERE Date = '20XX-10-11'`

Which of the following is the appropriate result of executing the SQL statement that is given below on the “Student” table and the “Department” table?

```
SELECT Name FROM Student, Department
WHERE Affiliation = DepartmentName AND Department.Address = 'Shinjuku'
```

Student

Name	Affiliation	Address
Tomoko Goda	Science	Shinjuku
Shunsuke Aoki	Engineering	Shibuya
Satoshi Kawauchi	Humanities	Shibuya
Yuko Sakaguchi	Economics	Shinjuku

Department

DepartmentName	Address
Science	Shinjuku
Engineering	Shinjuku
Humanities	Shibuya
Economics	Shibuya

a)

Name
Tomoko Goda

b)

Name
Tomoko Goda
Shunsuke Aoki

c)

Name
Tomoko Goda
Yuko Sakaguchi

d)

Name
Tomoko Goda
Shunsuke Aoki
Yuko Sakaguchi

Q10

Which of the following is the appropriate SQL statement for extracting Product information (ProductNumber, ProductName) of the products where InventoryQuantity is less than 100 units from the “Product” table and the “Inventory” table?

Product

ProductNumber	ProductName	UnitPrice

Inventory

ProductNumber	InventoryQuantity

a)
```
SELECT X.ProductNumber, ProductName FROM Product X, Inventory Y
  WHERE X.ProductNumber = Y.ProductNumber OR InventoryQuantity < 100
```

b)
```
SELECT X.ProductNumber, ProductName FROM Product X, Inventory Y
  WHERE InventoryQuantity < 100
```

c)
```
SELECT ProductNumber, ProductName FROM Product
  WHERE EXISTS (SELECT * FROM Inventory WHERE InventoryQuantity < 100)
```

d)
```
SELECT ProductNumber, ProductName FROM Product
  WHERE ProductNumber IN
        (SELECT ProductNumber FROM Inventory
                WHERE InventoryQuantity < 100)
```

Q11

Which of the following is the SQL statement for defining the derived table when the table of the relational database is accessed from the program by using embedded SQL statements?

a)	CLOSE statement	b)	DECLARE CURSOR statement
c)	FETCH statement	d)	OPEN statement

Q12

In the distributed database system, which of the following is a method that updates the database after inquiry is made for multiple sites that perform a series of transaction processing whether they can be updated or not and if it is confirmed that all sites can be updated?

a)	Two-phase commitment	b)	Data cleansing
c)	Data mining	d)	Replication