Software Development Technologies

Advanced Information and Communication Technology Training (Mandalay)

[Database Design and Administration]

Contents at a Glance

Table of Contents

1. Data Modeling

1.1. Introduction to Data Modeling

1.1.1. Information Systems Development and Data Modeling

Data modeling is a process to develop data model that describes how data is represented and used in information systems. The data modeling usually refers to designing the database structure using graphical representations such as entity-relationship modeling diagrams, however it is more important than just to develop data model but it is also important to capture actual requirements of information systems that can full fill the client needs for their business enhancement and improvement.

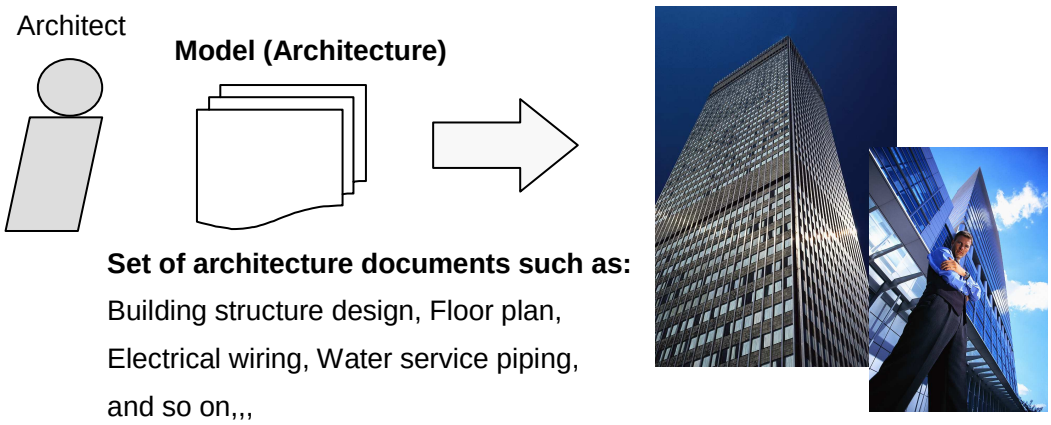


Figure 1 - What is Model?

As an example of modeling in general, an architect is a person who comes up with architecture, a building in this case, based on the client's requirements, and he/she needs to document down or models how the building should be constructed. Each document passes to responsible contractors to carry out the building construction in order to be completed it as what the architect modeled as shown in the figure above.

Now let us take a look at the same approach for information systems development. The information systems are also designed with modeling techniques as shown in below:

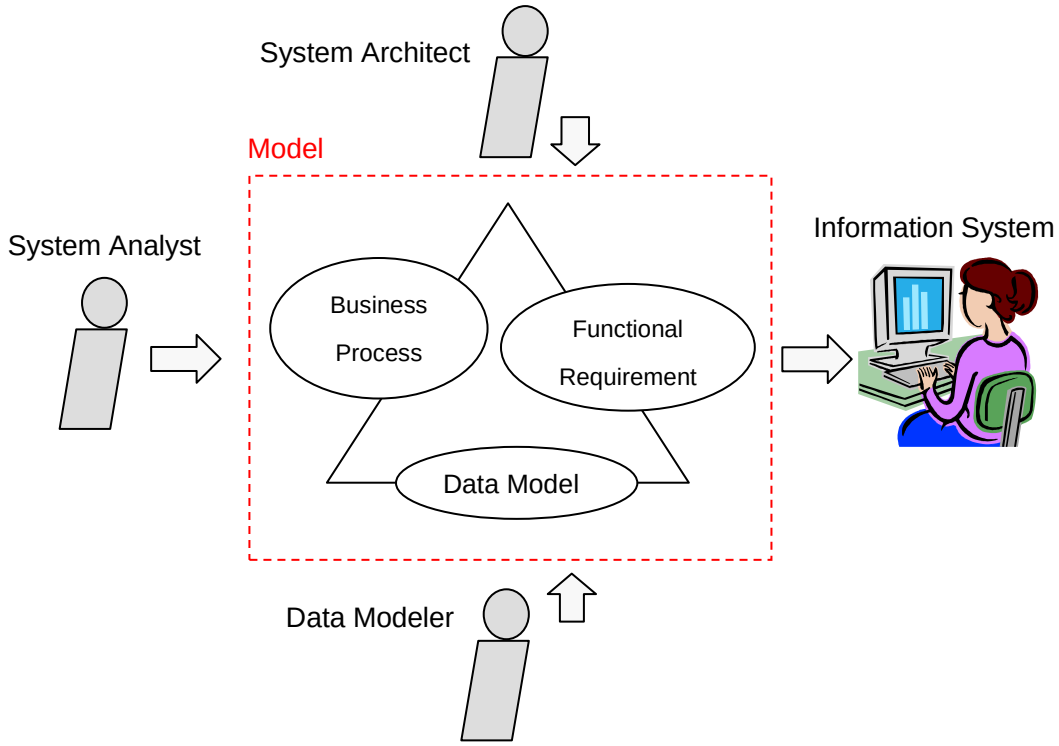


Figure 2 - Information System Model

The model for information system described with set of documents that outlined and described its structure, process and functional requirement and data model is a part of those design document that describes structure of the data.

1.1.2. Database Design Process

Database design process usually consists of three stages such as conceptual database design, logical database design and physical database design and as mentioned, data modeling or designing database is a part of information system development activities. The figure below depicts how data modeling and database design process involves with software development life cycle (SDLC). As can be seen, the conceptual database design is usually carried out during requirement analysis and definition phase, the logical database design in system design phase and the physical database design should be implemented during implementation phase:

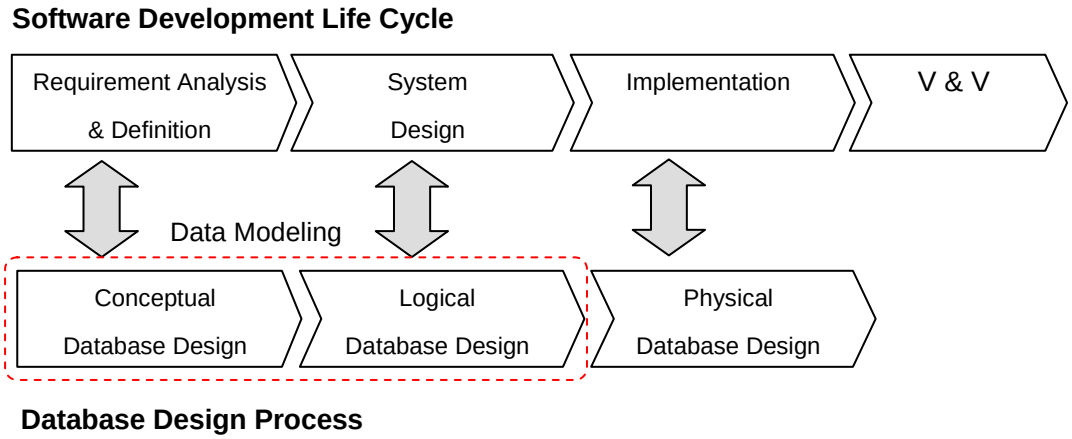


Figure 3 - Information System Development and Database Design Process

Another figure below is to look at what input takes those three database design processes and what are the outcome from each designing process. The conceptual database design needs overall information such as business model, organizational back ground and workflow in order to develop conceptual data model. The logical database design develops based on the conceptual data model together with processing and performance requirements to full file the end-user's need. Then the physical database design carried out to implement actual database system in development and production environment:

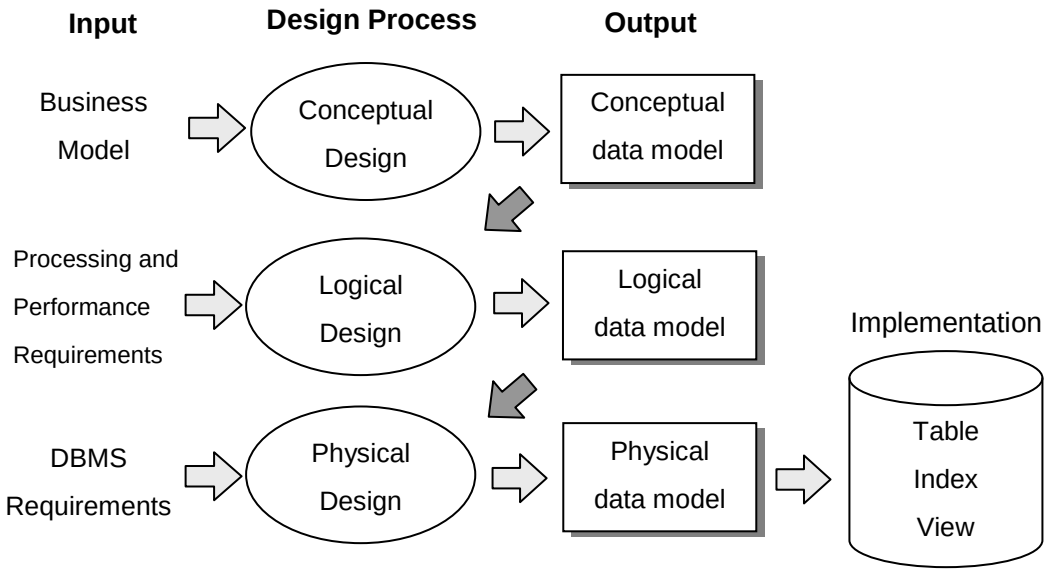


Figure 4 - Database Design Process

1.1.3. ANSI/SPARC Modeling Architecture

The figure below is ANSI/SPARC (Standards Planning and Requirements Committee) modeling architecture that was discussed in Fundamental Database. As we discussed earlier, the ANSI/SPARC modeling architecture consists of three schema such as conceptual schema, external schema and internal schema. The figure below overlaps data models with those three schema as follows:

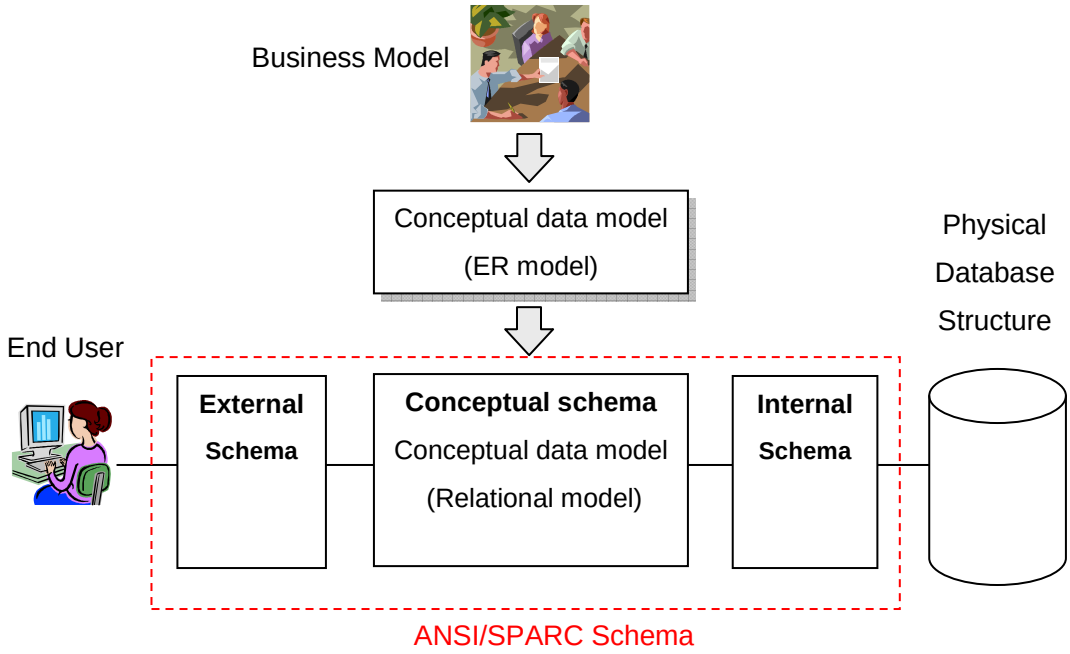


Figure 5 - ANSI/SPARC Schema Architecture and data modeling

The conceptual schema will be created based on conceptual data model described by ER diagram. The logical data model as conceptual schema must be independent form both external and internal schema. Table below is a list of those three schema and database design process:

Table 1 - ANSI/SPARC Schema and Database Design Process

ANSI/SPARC	Database Design Process	Description
Conceptual Schema	Conceptual Database Design Process	Overall data structure. System analyst's point of view.
External Schema	Logical Database Design Process	User interface data structure. End-user's point of view.
Internal Schema	Physical Database Design Process	Technical data structure. System administrator's point of view.

1.1.4. Conceptual Data Modeling Process

Data modeling process consists of two sub processes such as top-down data modeling and bottom-up data modeling. The top-down modeling develops outlined ER diagram as overall modeling. The bottom-up data modeling develops detailed ER diagram that is going to be a logical data base design. The data modeling process follows steps

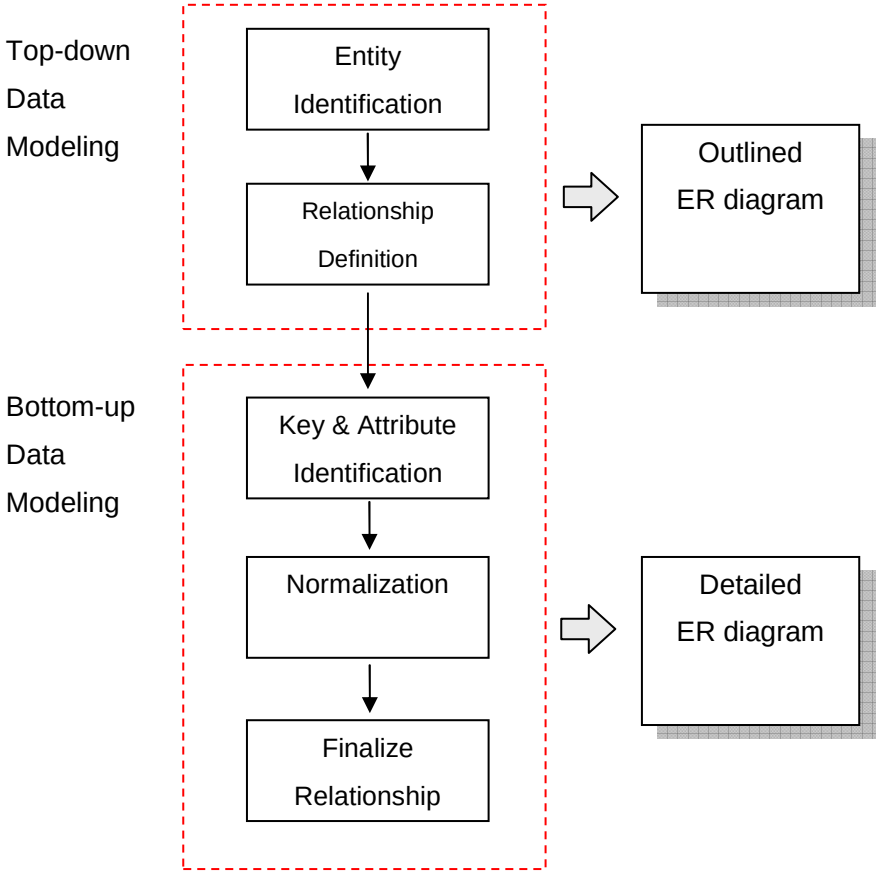


Figure 6 - Data Modeling Process

1.1.5. Data Modeling Diagrams

There are several ER diagrams for data modeling are available such as Information Engineering (IE or Crow's Feet), IDEF1X, BACHMAN and Chen modeling. Following table and figure are details of each diagram. From the table below, IE and IDEF1X are most commonly used nowadays is because popular ER modeling available today tools such as ERWin, Visio are supporting those diagrams.

Table 2 - Data Modeling Diagrams (ER diagrams)

Diagram	Description
IE	Information Engineering (Crow's Feet)
IDEF1X	IDEF is the family of ICAM Definition Languages. IDEF1X is for data modeling.
BACHMAN	Charles Bachman's Model
Chen	Peter Chen's Model

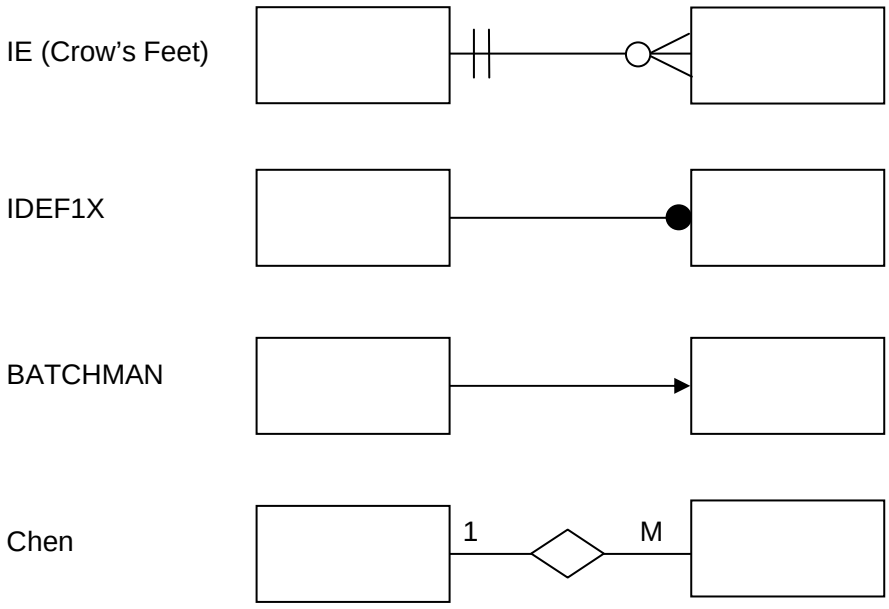


Figure 7 - Data Modeling Diagram (ER diagram) notations

1.2. Top-down Data Modeling

There are basically two types of approaches of data modeling such as top-down data modeling and bottom-up data modeling. As for the top-down data modeling is the approach to model the data from organizational point of view. Based on the business model and its workflow, following modeling processes will be carried out in order to come up with the conceptual database design.

1.2.1. Entity Identification

Let us first of all, quick review the elements of ER diagram as we studied during Fundamental Database such as entity, attribute and relationship with associated database elements shown in table below:

Table 3 - ER diagram and Database

ER diagram elements	Database elements	Description
Entity	Table	Entities as database tables
Relationship	Business Rule	Business Rules such as constraint, trigger
Attribute	Column	Columns in the table
Identifier	Primary key	Keys such as Natural key and Surrogate key

In order to identify entities from business model, understanding the type of entity is important. As can be seen table below, there are three types of entities:

Table 4 - Type of entities

Type	Description	Examples
Resource Entity	Physical such as person	Customer, Employee, Member
	Physical such as object	Product, Material, Parts
	Physical such as place	Office, Meeting room, Warehouse
	Conceptual	Organization, Department, Team
Event Entity	Transactional	Quote, Sales, Purchase, Stock
Summary Entity	Analytical	Sales report by product, by time frame

Resource entity is static entity such as customer, employee to be referred by event entity to provide master information. Event entity is dynamically update during transactional processing such as order and order items. Summary entity is to summarize transaction data in event entity such as monthly, weekly sales reports. It is important in this stage of the process to identify and categorize those entities for subsequent process.

1.2.2. Relationship Definition

After identifying and categorizing entities, next step is to glue those entities with business rules. This process is called relationship definition. Set of figures below are sample ER diagrams based on identified entities by type of relationship using IE and IDEF1X notations:

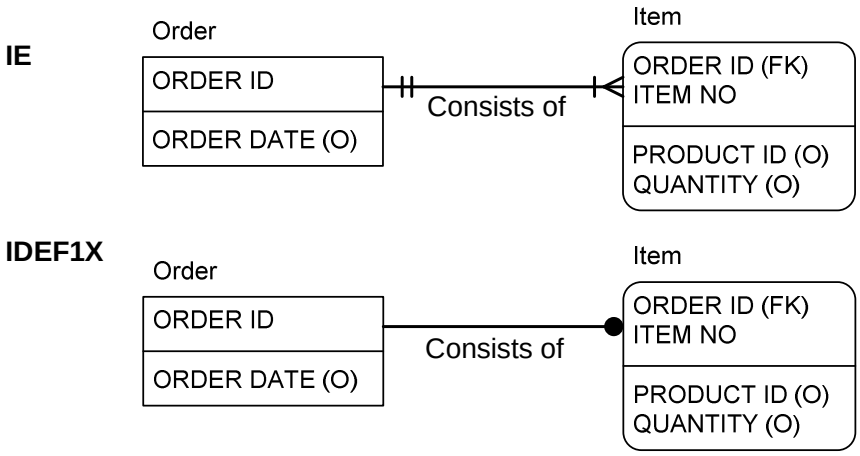


Figure 8 – One-to-Many (Mandatory, Mandatory)

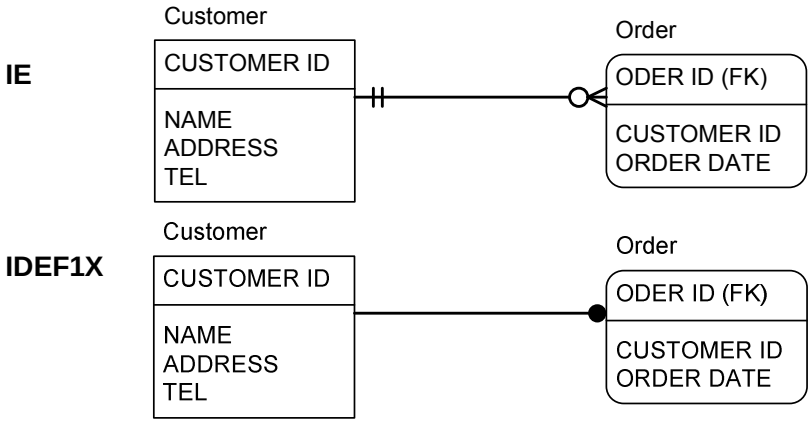


Figure 9 – One-to-Many (Mandatory, Optional)

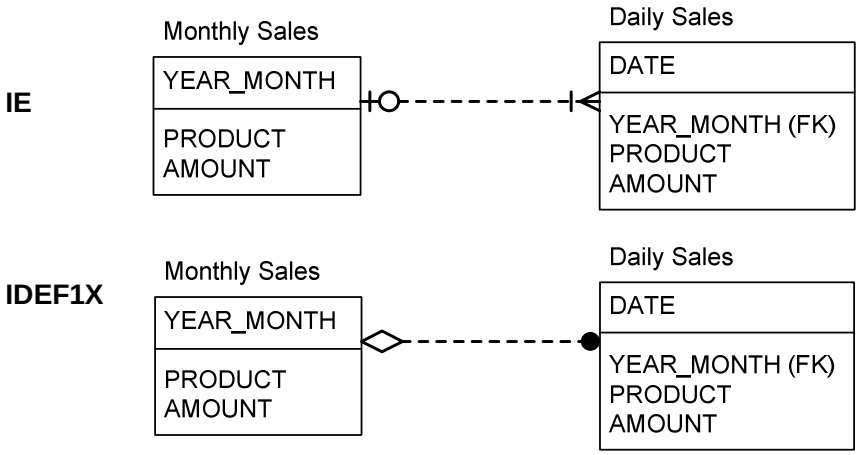


Figure 10 – One-to-Many (Optional, Mandatory)

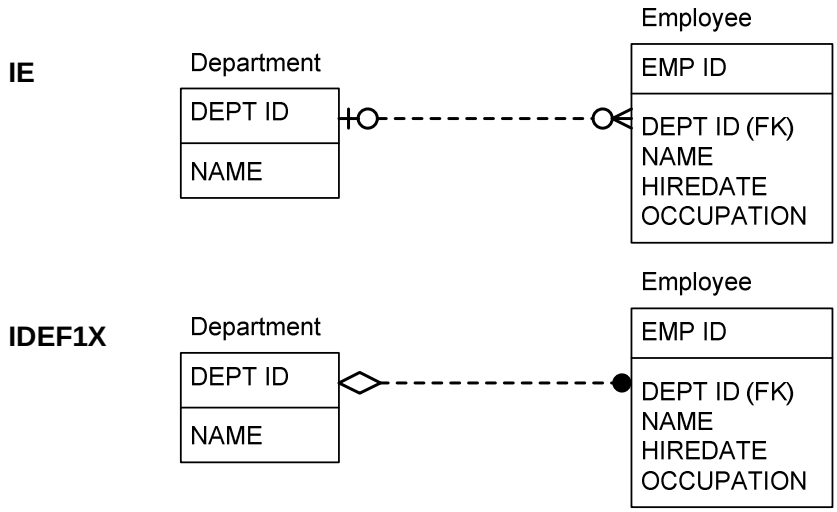


Figure 11 – One-to-Many (Optional, Optional)

Many-to-many relationships can be identified during the data modeling stage. Conceptually, the data model of the many-to-many relationship consists of two entities however in actual database designing stage, those two entities need to have another entity in between it is called as cross-entry (or join-table). The cross-entity is the joined table to resolve the many-to-many relationship into two "one-to-many" relationships. The primary key of the cross-entity is normally composite of the primary key from each entity. Figure 12 is an example of the many-to-many relationship of Lecturer and Subject. In this example, the primary key of the cross-entity (Lectuer_Subject) is compound of lecturer_id and subject_id as can been seen in the figure.

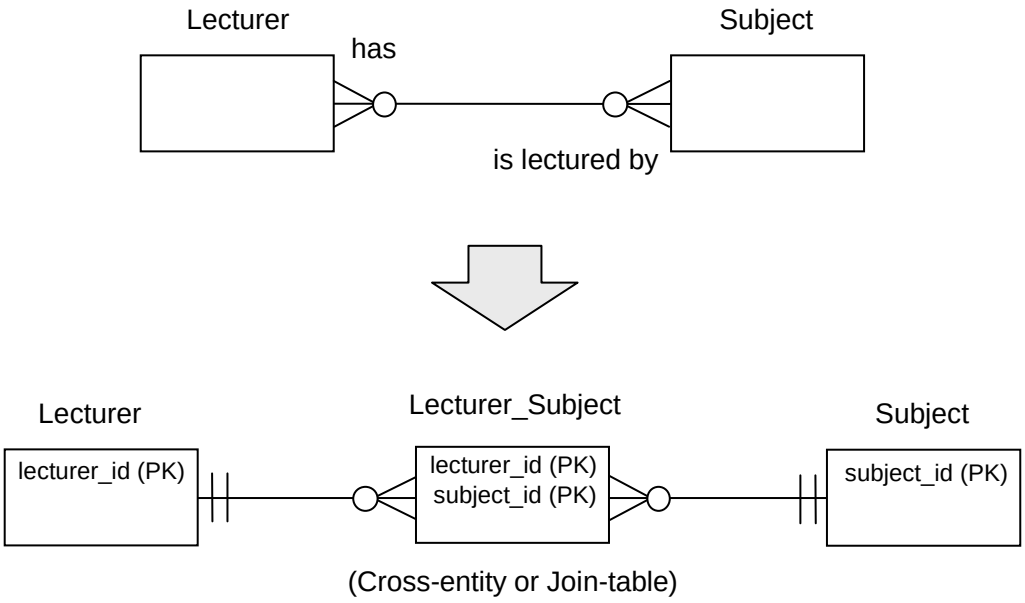


Figure 12 – Many-to-Many Relationship

Apart from relationship between two entities, another relationship is about categorizing entity. This relationship is something like inheritance in UML diagram as can be seen figures below. The relationship is in between parent and child/children entity:

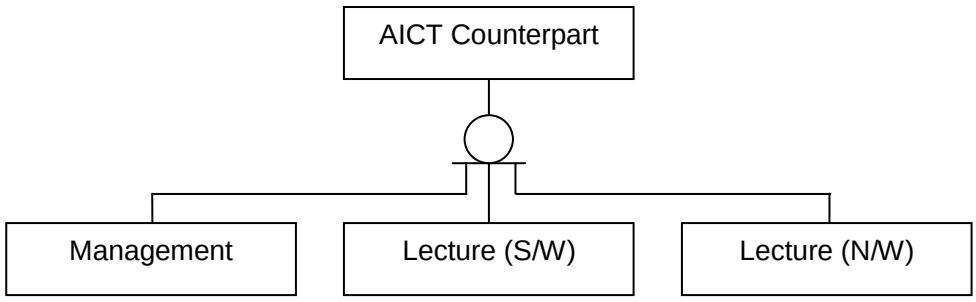


Figure 13 - Category of entity

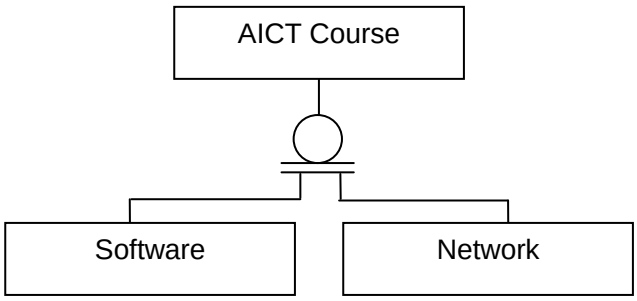


Figure 14 - Category of entity

1.2.3. Attribute and Key Identification

After identified entities and create relationship in between, next process is to identify attributes for each entity that can be carried out as second last step of the conceptual data modeling process. Some of attributes can be identified from business models and interviews from management of the organization during requirement analysis and definition phase, however details of attributes can be identified during bottom-up modeling stage later on with other activities such as interviewing end-user and analysis existing business documents. Once attributes are identified, then analyze what would be a primary key of each entity. Tips of how to find out primary key is to review identified attributes with following points:

- What are the constant value of the attribute
- Any possibility of the attribute with Not Null situation
- What are the minimum combination of attributes as a composite key

Following figure is a sample process of identifying primary key from attributes:

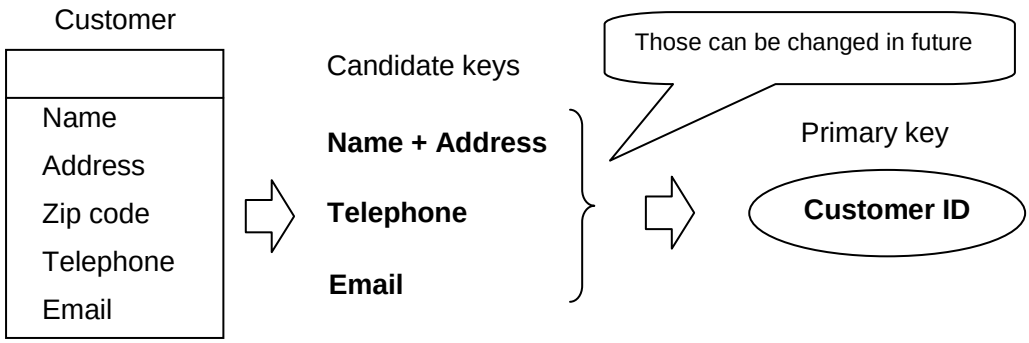
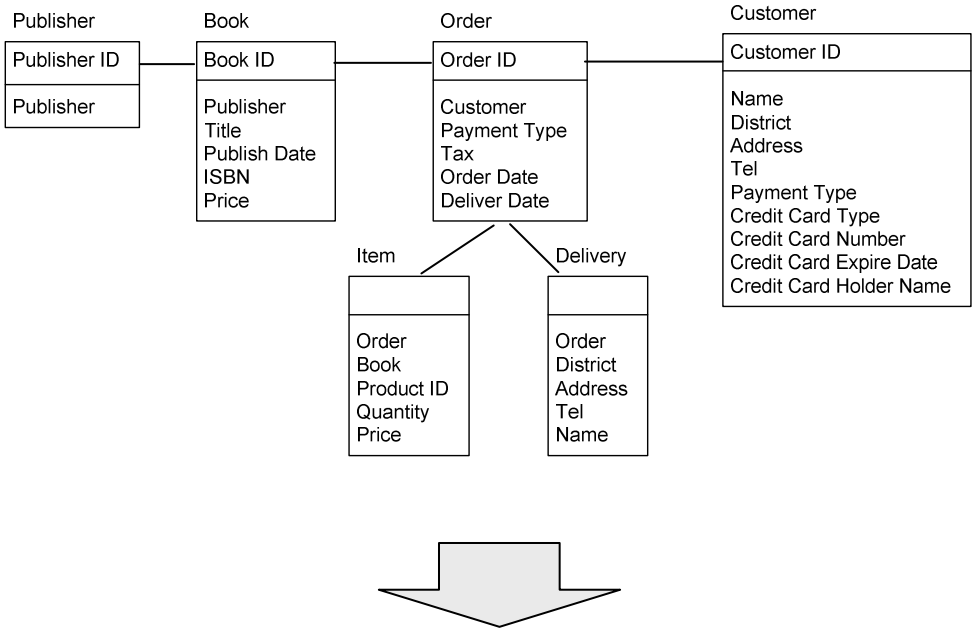


Figure 15 - Find out attribute and key of the entity

1.2.4. Finalize Relationship

Finally, it is necessary to conduct complete review of the ER diagram you came up with during the top-down conceptual database design process in order to verify each relationship and entity and primary/foreign keys. Figure below is a sample conceptual data model that describes a part of bookshop ordering system. As can be seen, outlined data model (Figure above) is still outlined because type of relationships, primary/foreign keys and dependencies are not clearly identified yet compare with Finalized data model (Figure below):

Outlined data model



Finalized data model

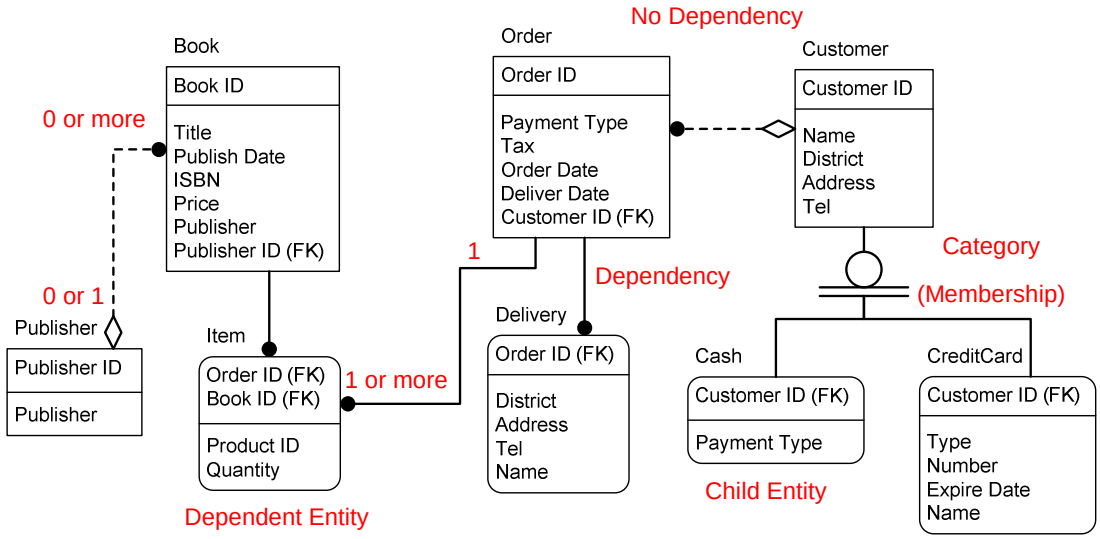


Figure 16 - Sample ER diagram final review

1.3. Bottom-up Data Modeling

Bottom-up data modeling is another approach to model the data from end-user's point of view. Based on the interview from system and end users and analyze current or existing systems if they have and collect business documents to understand the structure of information. Usually following bottom-up modeling process will be carried out in order to come up with another angle of conceptual database design:

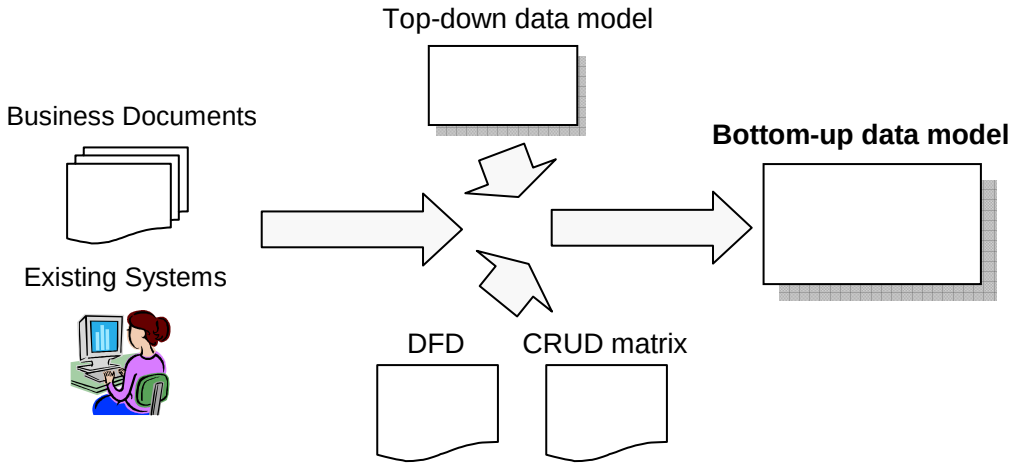


Figure 17 - Bottom-up Modeling process

1.3.1. Business Documents

First step of bottom-up modeling is to collect currently used business documents in the office such as sales order, purchase order, invoice and so on, and existing system screen shots with real business data. Those information are starting point to identify attributes and keys in order to map and verify with the top-down data model conducted before.

1.3.2. DFD

In order to verify business work flow and conceptual data model, to draw a Data Flow Diagram is also important to map association of business events and entities. Details of DFD was discussed in Fundamental Application Development.

1.3.3. CRUD Matrix

CRUD stands for Create, Refer, Update and Delete is four basic functions of persistent storage and it is used for bottom-up data modeling to analyze when and what events happens to each entity and its attribute.

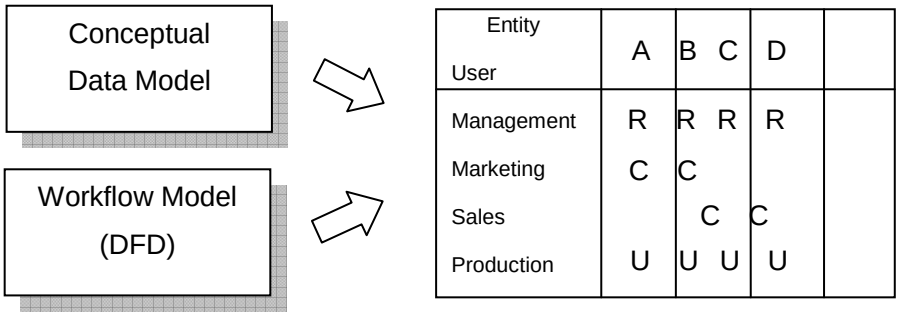


Figure 18 - CRUD matrix

1.3.4. Normalization

Normalization process must be carried out in order to get rid of redundancy from the data model. This guarantees the concept of DOA (One fact in one place). Normalization process and technique were discussed in Fundamental Database. Please refer it for details.

1.4. Integration of the modeling

1.4.1. Data Model Verification

Both the top-down model and the bottom-up model finally need to be integrated as the result of the conceptual data model or conceptual database design. As can be seen table below, it is necessary to conduct comparison of those two modeling then it should be reflected in final conceptual data model ER diagram:

Table 5 - Sample integration of Top-down and Bottom-up modeling

Top-down model Entity (Primary Key)	Bottom-up model Entity (Primary Key)	Integrated model Entity (Primary Key)
Customer (Email)	Customer (Customer ID)	Customer (Customer ID)
Employee (ID Card Number)	Employee (Employee ID)	Employee (Employee ID)
Head office (Address)		Department (Department ID)
Book (ISBN)	Book (Product ID) Magazine (Product ID)	Product (Product ID)
	Credit Card Member (Card Type, Card Number)	Credit Card Member (Card Type, Card Number)

1.5. Sample Data Modeling by business application module

Data models discussed in this section are based on the following sample business workflow. Each data model is for business application module such as Quotation, Sales Order, Purchase Order, Receipts of Goods, Dispatch Advise and Invoice. (Accounting is outside of this modeling scope)

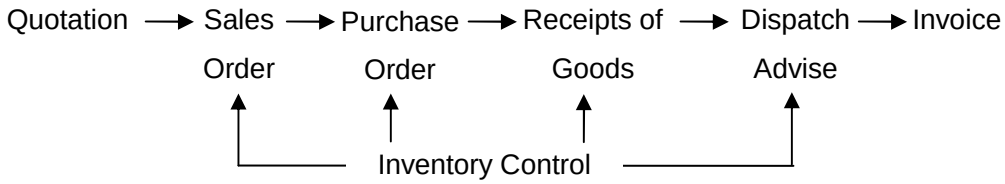


Figure 19 - Business workflow for sample data model

1.5.1. Product Catalog

Definition of product is very much depend on the business nature of the company. Product can be item, row material, property, perishable foods, service or some others. Example here is about product catalog for apparel company who sells women's and men's cloths. As can be seen product catalog may have resource entities such as category, size, color, season of the product to classify products:

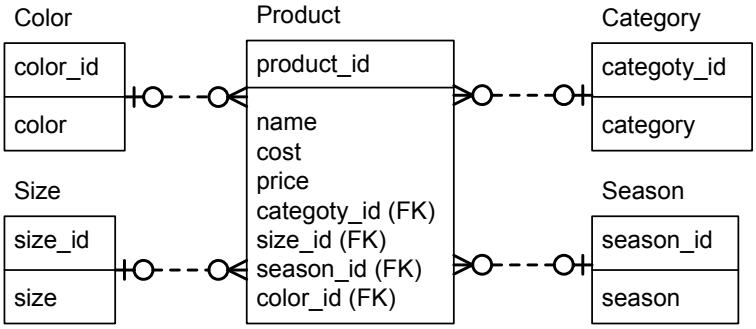


Figure 20 - Product Catalog

1.5.2. Quotation and Sales Order

Sale order will be generated against quotation in this model. Thus quotation and sales order usually has one-to-many relationship. Actual quotation is normally revised several times till business deal is closed hence Quotation ID will be designed as combination of quotation ID + revision number such as 000001-01 (000001 is Quotation ID and 01 is revision number of the quotation):

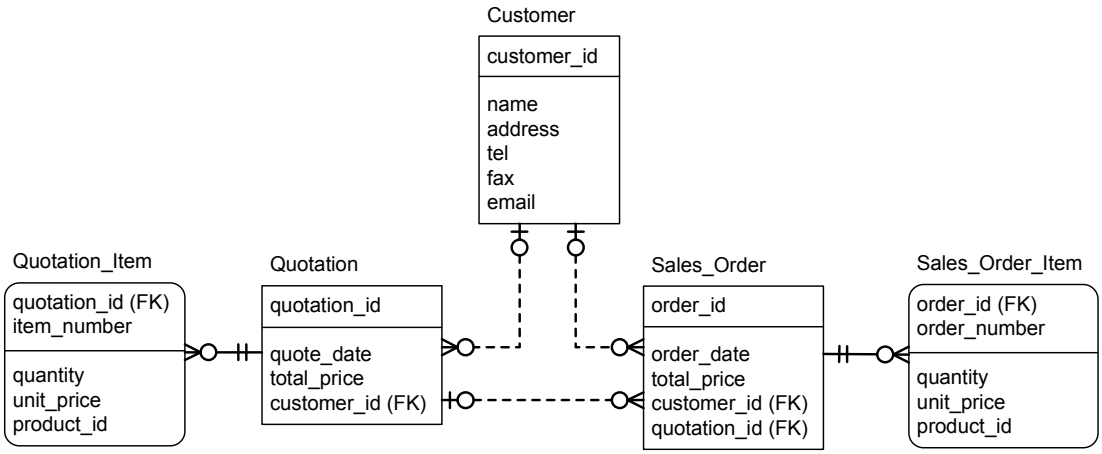


Figure 21 - Quotation and Sales

1.5.3. Purchase Order

Data model for purchase order consists of variety kind of types as well. Especially for manufacturing is complicated because of the involvement of JIT (Just-In-Time) or SCM (Supply Chain Management) concepts behind. Figure below is a simple data model for distribution industry:

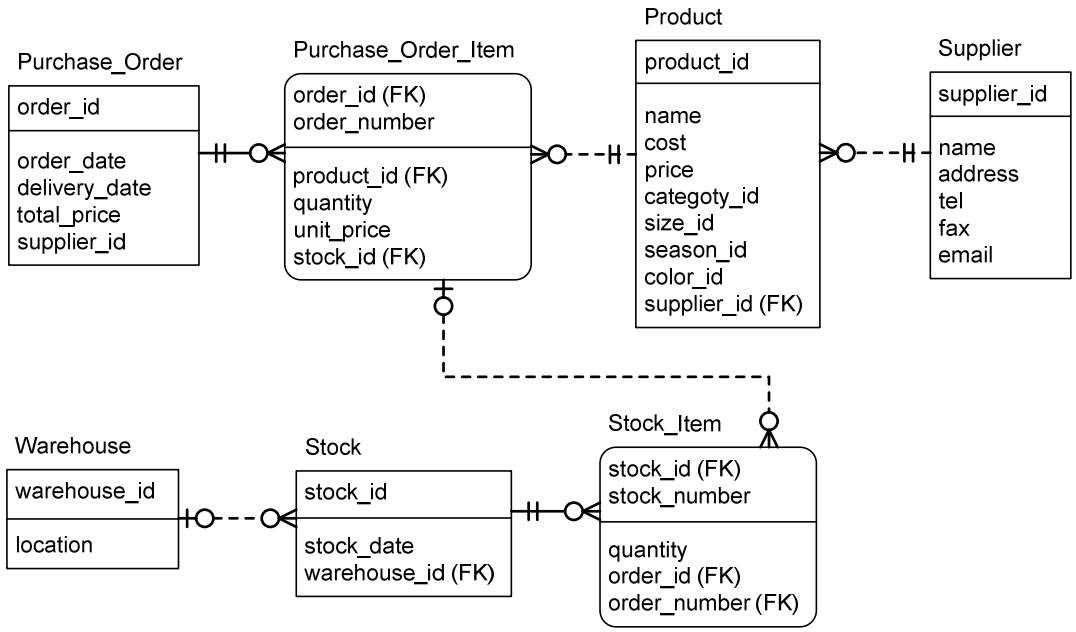


Figure 22 - Purchase Order

1.5.4. Inventory Control and Dispatch Advise

Inventory control is management technique to minimize the total cost of inventory with concerning of safety level of the stock. It is also many approaches to manage stocks depend on business models. Figure below is a simple inventory control data model just controlled by number of stocks by products:

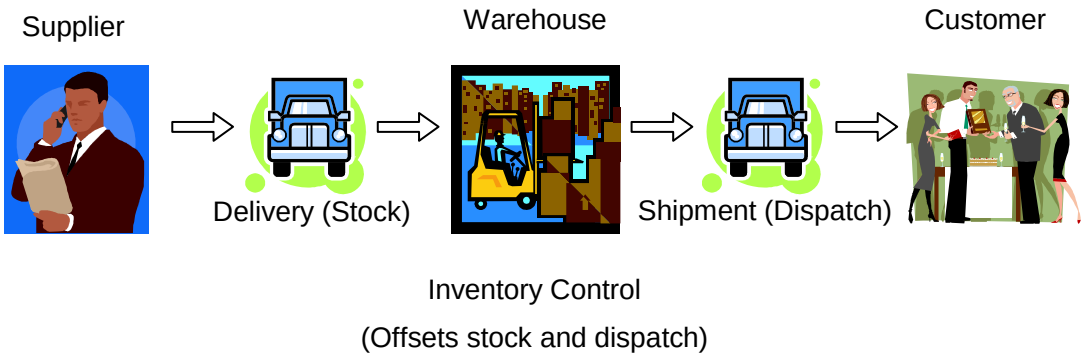


Figure 23 - Inventory control overview

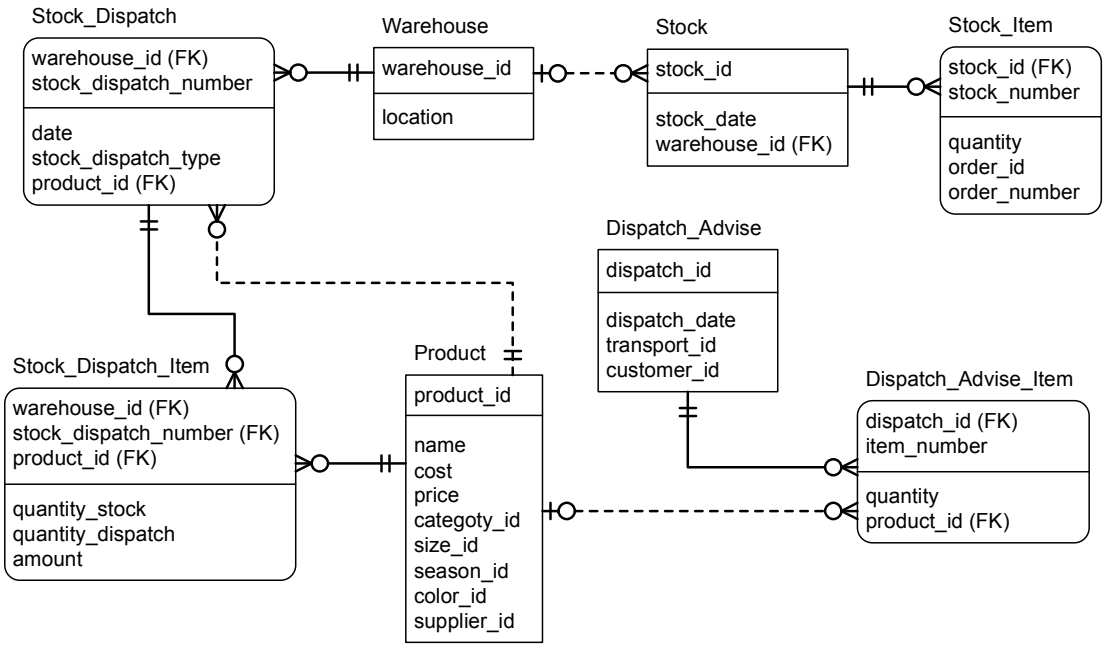


Figure 24 - Inventory Control and Dispatch Advise

1.5.5. Invoice

Invoice has a relationship with sales order. The relationship might be one-to-one but it could be one-to-many sometimes because of partial invoice. Some cases invoice will be sent to customer as bulk invoice according to the business contract such as every end of the month, however figure below is a simple invoice data model for individual invoice:

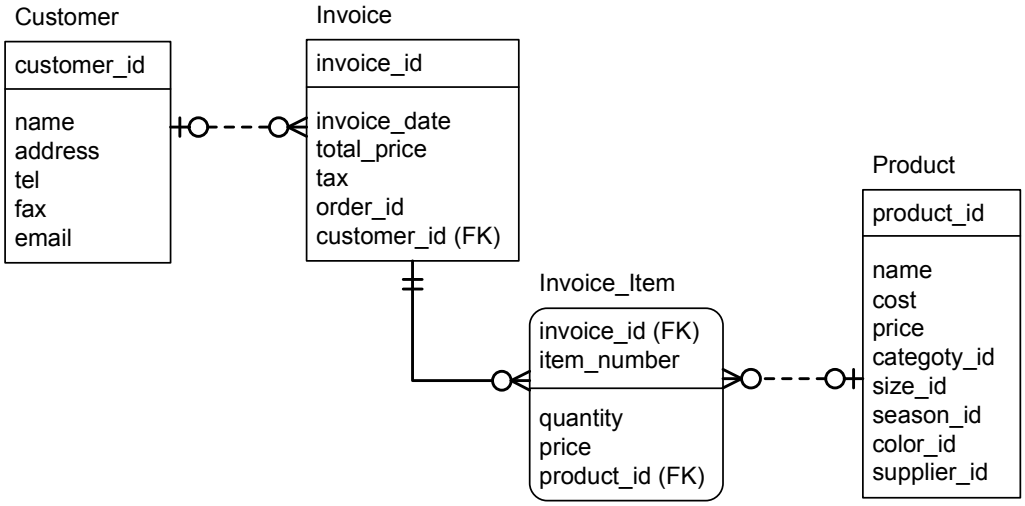


Figure 25 - Invoice

Exercise 1 - Data Modeling practice with case study

1. Develop a conceptual data model for following online bookshop system on the Internet

1-1. Background information

A joint venture of "Yangon eBooks" is a project of establishing an online bookshop to sell their books on the Internet. This ongoing project is expected to start the business from 2008. Member customers of the bookshop in Myanmar can order books and settle their payment by their credit card. The area of delivery covers only domestic and limited city area at the beginning of the project.

According to initial business planning, the bookshop operation will start with their stocks of 100,000 books and estimated with 10,000 initial members. Yearly 100% of growth of membership is projected for next 3 years. About 10,000 orders per month are expected in first year. The bookshop deals a contract with domestic Logistic Service Provider (LSP) who is responsible to deliver books to the member customers.

"Yangon eBooks" requires their information system which consists of two applications such as (1) Online bookshop Web application for customer to order and (2) Back-office application for supporting their business. Details of those two applications are described in 1-2. to 1-9.

1-2. Business model overview

Figure below depicts overview of Yangon eBooks business model:

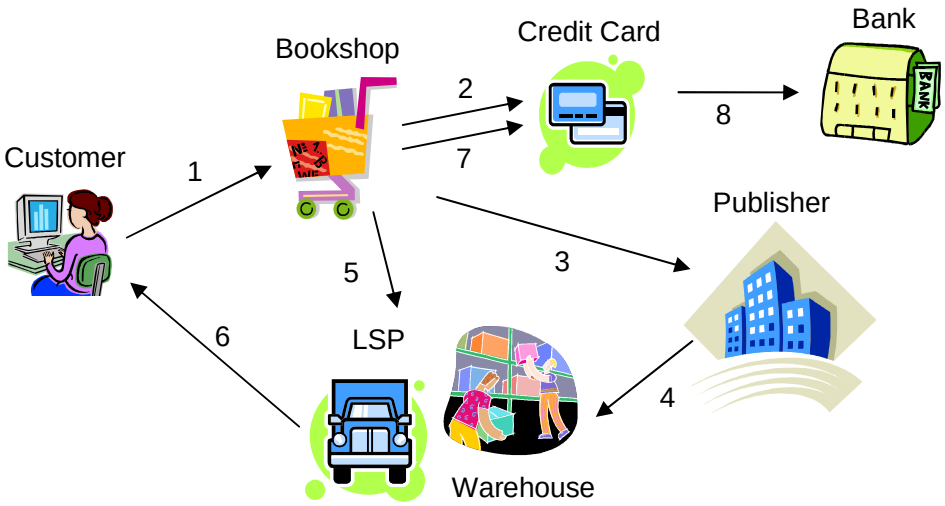


Figure 26 - Yangon eBooks business model

Table 6 - Yangon eBooks business model

No	Workflow	Description
1	Confirm Order	Customer orders book(s)
2	Verify Credit Card	Verify customer's credit card (Authorization of the credit card)
3	Purchase Order to Publisher	Bookshop places purchase order to publisher
4	Receipt of goods at Warehouse	Warehouse receipts the goods (books) from publisher
5	Dispatch Advice to LSP	Bookshop advises LSP for dispatch of the books to customer
6	Deliver book(s) to Customer	LSP delivers (transports) book(s) to customer
7	Invoice to Credit Card Company	Bookshop sends invoice to credit card company
8	Direct Debit (Account Transfer)	Credit card company debits customer's payment from bank

Note: LSP (Logistic Service Provider)

1-3. Workflow

Figure below depicts overview of Yangon eBooks business model using workflow diagram:

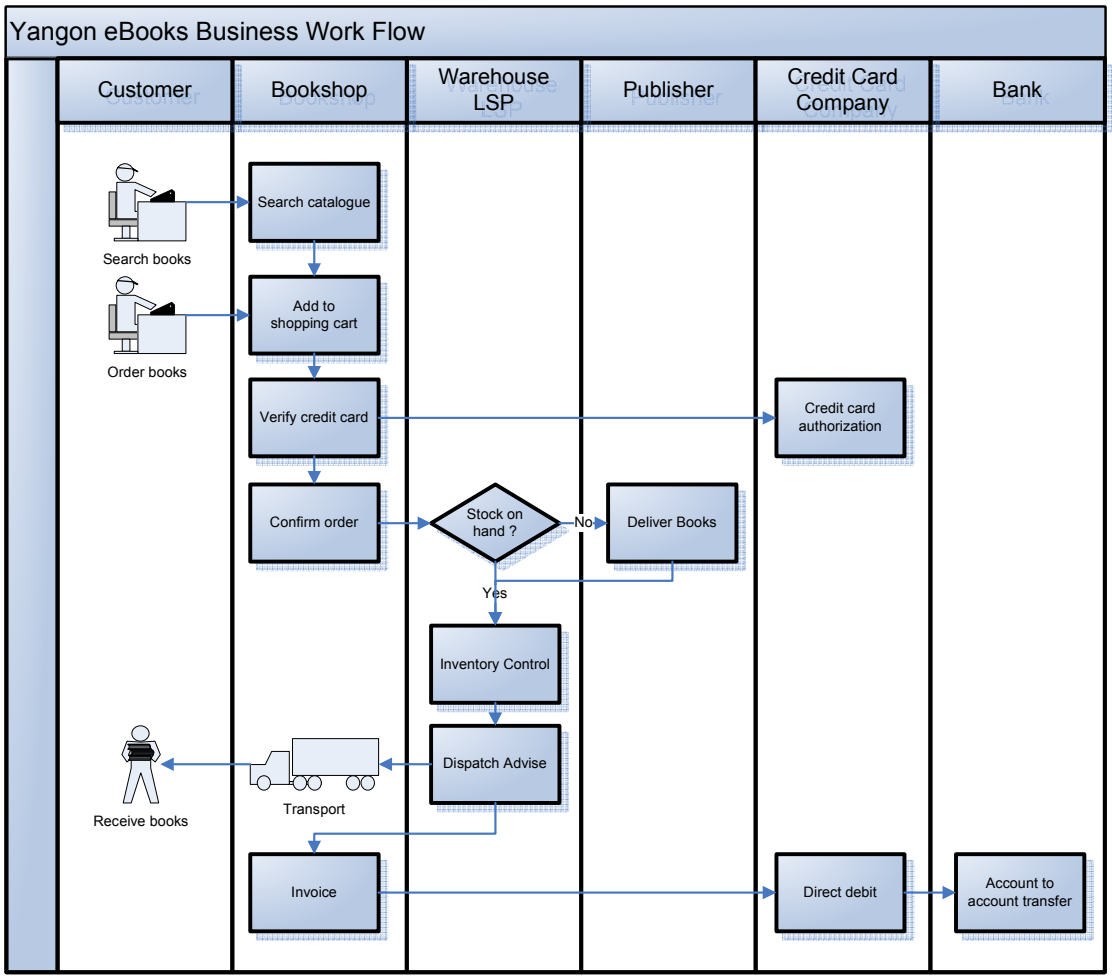


Figure 27 - Yangon eBooks Workflow

1-4. Organization Chart

Chart below is structure of Yangon eBooks organization:

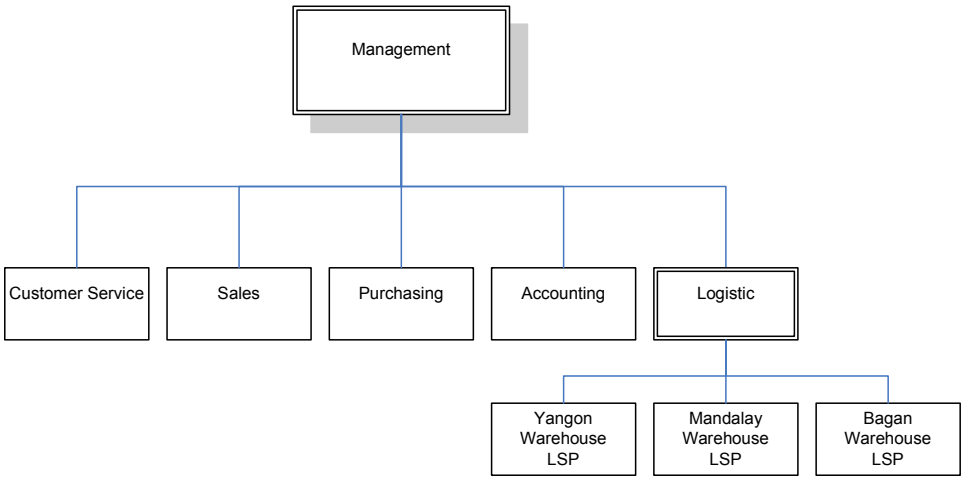
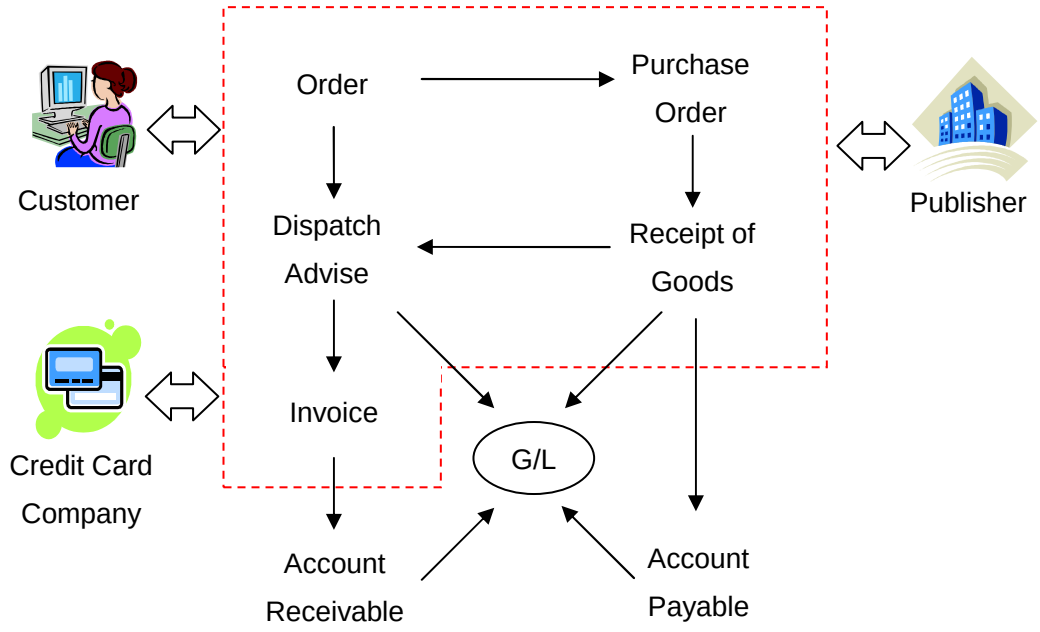


Figure 28 - Organization Chart

1-5. Scope of the system and data modeling

Figure below is an overview of the information system for "Yangon eBooks" and dotted line in red is our scope of this exercise. It describes interrelationship of system modules:



Note: G/L (General Ledger)

Figure 29 - Scope of the system and data modeling

1-6. Business Rules

In order to make modeling as simple as possible, followings are caps to restrict business rules:

- Cancellation and change of order after confirmation of the order dose not allowed. (No cancel, no change policy)
- Confirmed order can be delivered when all the items are stocked in the warehouse. (No partial delivery is accepted)
- Ordered books delver to customer's address. (Delivery address must be fixed)

1-7. Business Documents

Table below is definition of business documents for this data modeling practice:

Table 7 - Business Documents

No	Document	Description
1	Order	Order for customer to order books
2	Purchase Order	Purchase Order for bookshop to order books from publisher
3	Receipt of Goods	Receipt of Goods for warehouse to control inventory
4	Dispatch Advise	Dispatch Advise for LSP to deliver books to customer
5	Invoice	Invoice for bookshop to credit card company

1-8. Data Flow Diagram

Figures below are Data Flow Diagrams (DFD) such as Context Diagram and 1st Level Diagram of Yangon eBooks Web site:

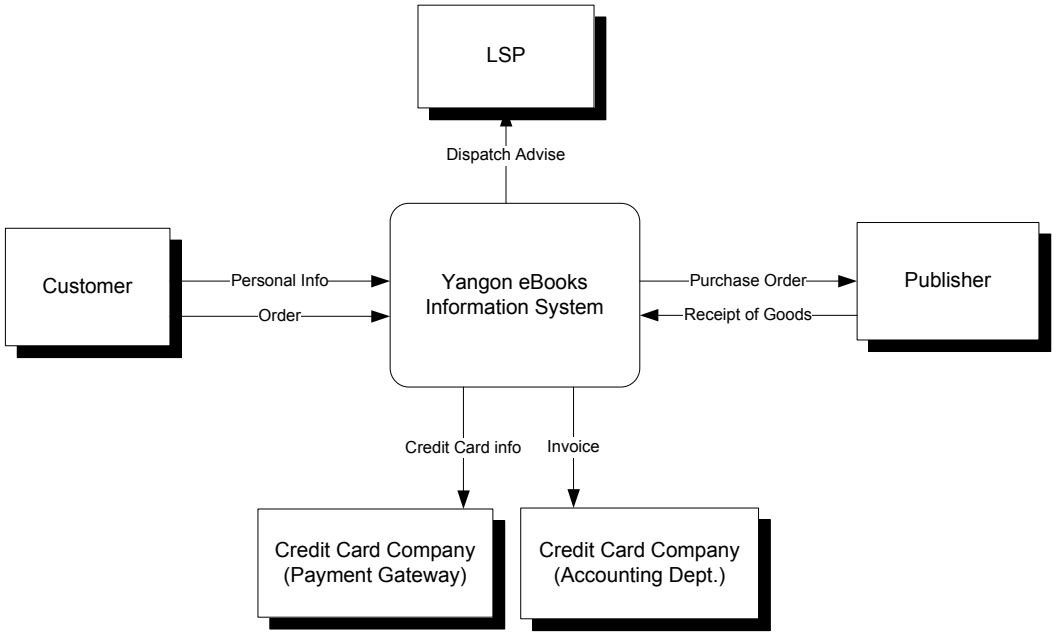


Figure 30 - Yangon eBooks DFD - Context Diagram

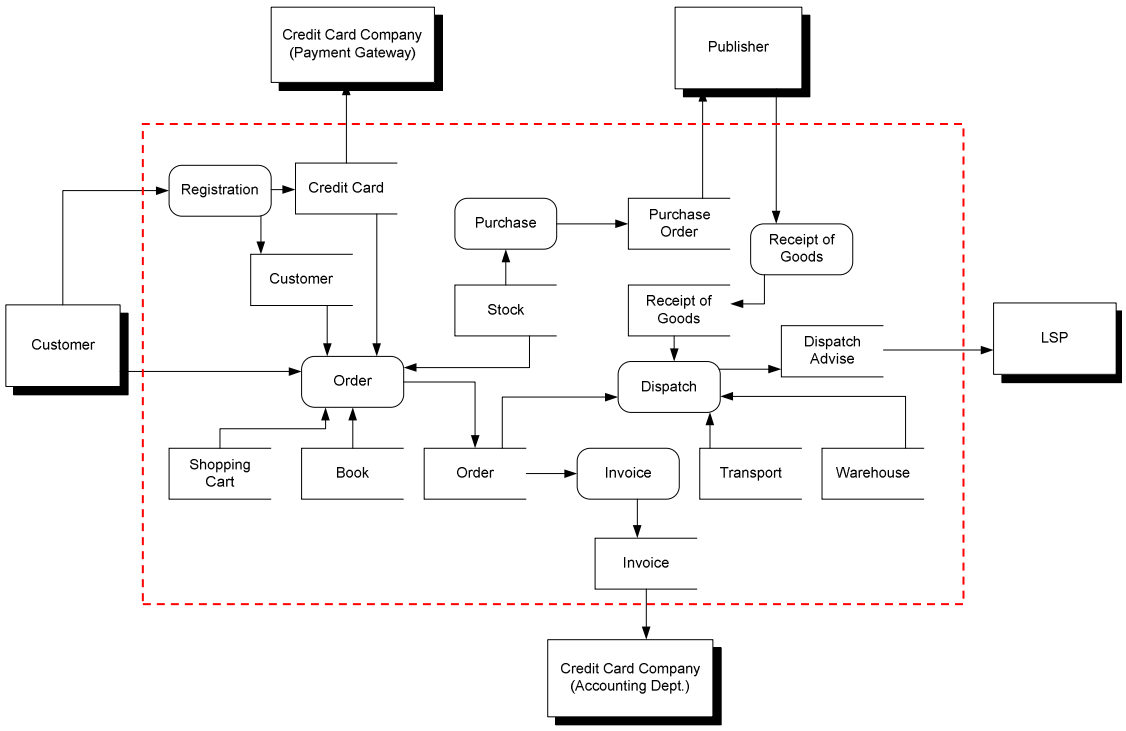


Figure 31 - Yangon eBooks DFD - Level 1 Diagram

1-9. User interface

Figure below is user interface transition of Yangon eBooks Web site for bottom-up data modeling:

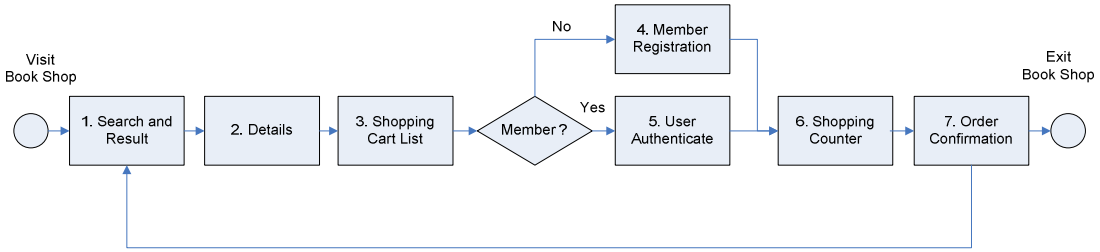


Figure 32 - Yangon eBooks User Interface Transition

Table below is description of each user interface described in figure above:

Table 8 - Yangon eBooks User Interface List

No	Page Name	Description
1	Book Search and Result	Customer can search with their search criteria such as category of the book, title and author of the book. The search result page will be shown upon clicking of the “Search” button
2	Details	Details information of the book such as title, author, publisher, date of issue, price and customer’s comments. Customer can add the to shopping cart..
3	Shopping Cart List	Customer can view list of books, price, quantity, and total price he/she puts in the shopping cart.
4	Member Registration	New customer can register with his/her personal particulars such as name, gender, date of birth, address, zip-code, telephone, email, and credit card information (Card name, Card number, Expire Date)
5	User Authenticate	Member can login to payment pages with his/her identification such as Member ID, email, and password.
6	Shopping Counter	Customer can finally confirm his/her order to be delivered. All the payment information such as price of each book, total price, taxes, shipping charge, payment methods will be shown.
7	Order Confirmation	Customer can view the order ID with his/her order. Email will be sent to the customer to inform confirmation of the order.

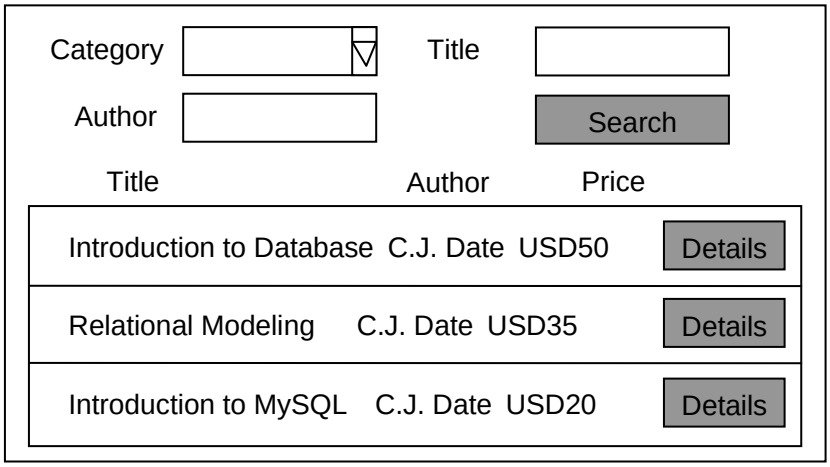


Figure 33 - Screen image: 1. Book Search and Result

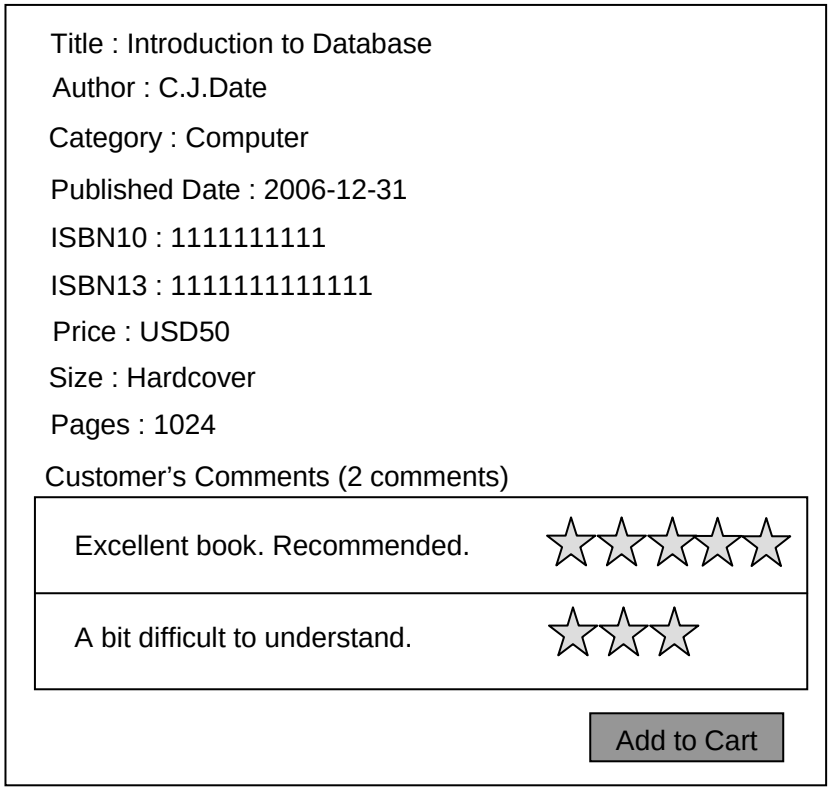


Figure 34 - Screen image:2. Details

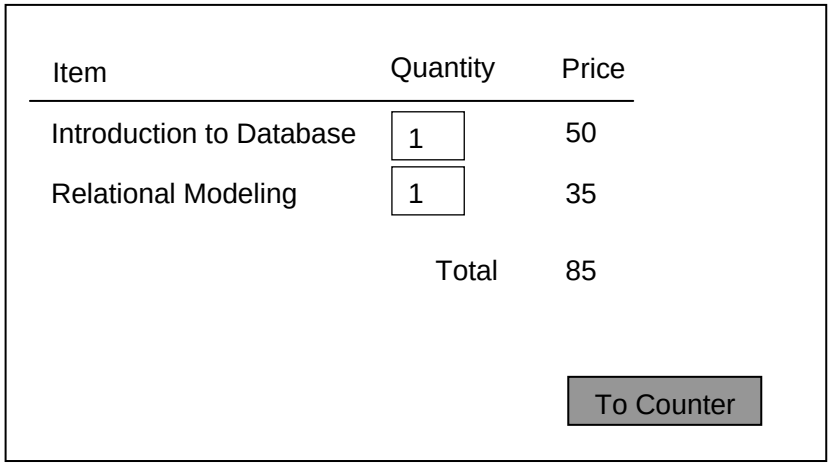


Figure 35 - Screen image:3. Shopping Cart List

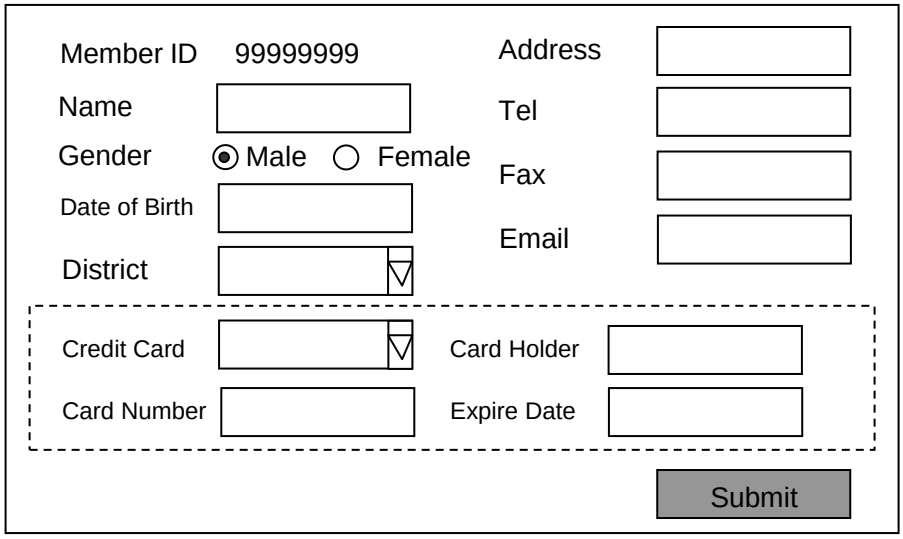


Figure 36 - Screen image:4. Member Registration

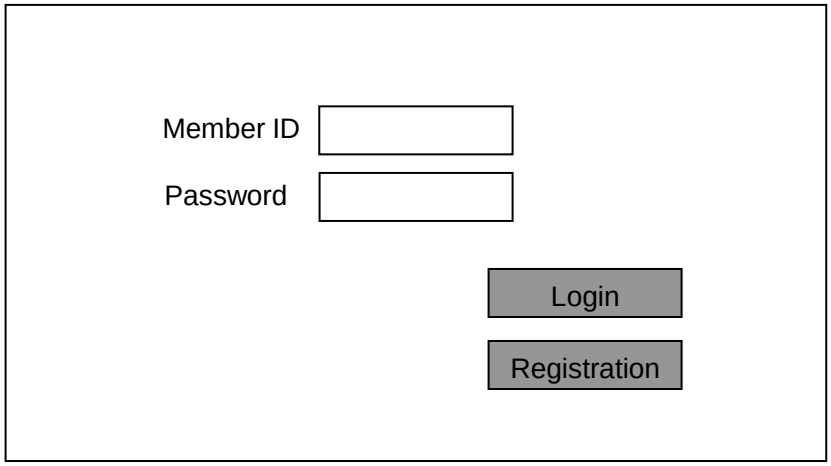


Figure 37 - Screen image:5. User Authenticate

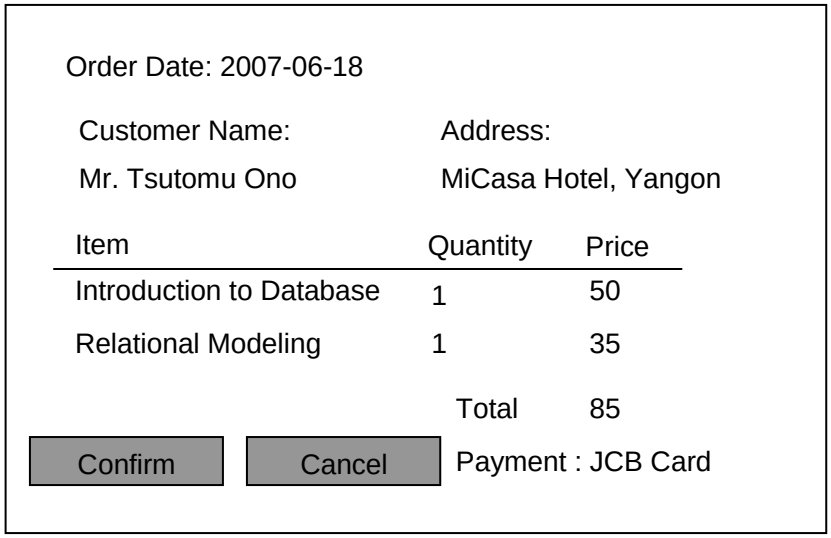
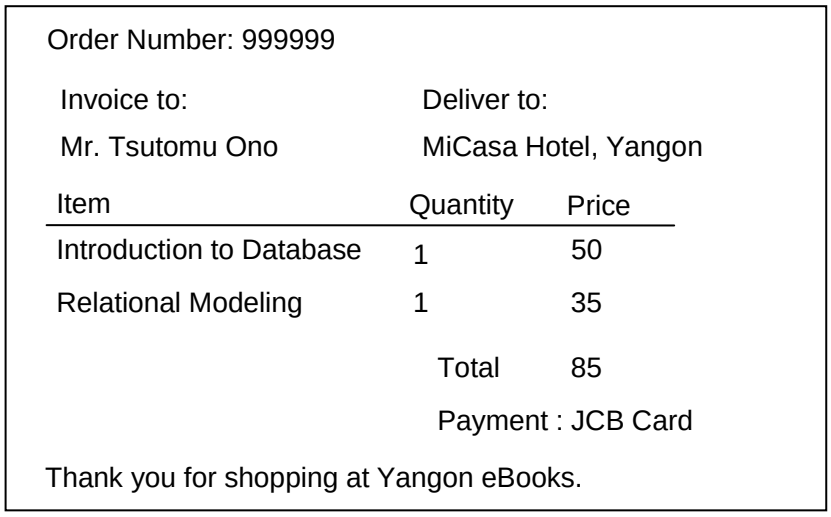


Figure 38 - Screen image:6. Shopping Counter



Order Number: 999999

Invoice to:	Deliver to:
Mr. Tsutomu Ono	MiCasa Hotel, Yangon

Item	Quantity	Price
Introduction to Database	1	50
Relational Modeling	1	35
	Total	85
	Payment : JCB Card	

Thank you for shopping at Yangon eBooks.

Figure 39 - Screen image:7. Order Confirmation

Note:

Please make sure to understand business model properly given to you. If not, please enquire to clear all your concerns with your lecture in order to carry out this conceptual data modeling exercise. At the end of the exercise, please submit your design document to your lecture for review and feedback to improve your data modeling skills.

2. Database Design

2.1. Logical Database Design

2.1.1. Introduction to Logical Database Design

Logical database design is a process to carried out followed by the conceptual database design that to describe logical structure of the database. The logical database design should take business requirements such as system response time of the application and database and business transactions into account in order to be designed it as performance considered database. The logical database design also needs to describe the indexing, data distribution and de-normalization from normalized conceptual database design, in case if the performance would be a bottleneck of the database. Following table lists down process of the logical database design phase:

Table 9 - Logical Database Design Process

No	Process	Description
1	Data volume estimation	Estimate database capacity described in table below
2	Business requirement consideration	Consider user's requirements and business transactions
3	Database Optimization	Optimize design based on business requirements above
4	View Design	Design view as external model for end-user's view point
5	Access Privilege Design	Design database security control given by limited access
6	ER diagram to RDBMS Mapping	Prepare for the physical database design as next step

2.1.2. Capacity Planning

Capacity planning is the process of adjusting the capacity of database to response to changing or predicted demands. It also estimates the size and structure of the database based on the data volume, its growth rate and period to store in the database by category of entities. Table below is a sample items that need to be included in the capacity planning:

Table 10 - Capacity Planning

Item	Description
Data Volume (Initial)	Initial Data Volume (Record length * Number of records)
Data Volume (Average)	Average Data Volume (Record length * Number of records)
Maximum Data Volume	Maximum Data Volume (Record length * Number of records)

Growth Rate	Growth rate of data volume (by day, week, month, quarter, year)
Preservation Period	Period of keeping data or event to delete it from database

Table below is an example of capacity planning by entities. As can be seen, data volume and growth rate of resource entities such as department and employee are stable, however event entities such as Order and Order Item are growth rapidly:

Table 11 - Sample Capacity Planning by Entity

Entity	Initial Data Volume	Average Data Volume	Maximum Data Volume	Growth Rate	Preservation Period
Department	50	50	100		N/A
Employee	10,000	10,000	50,000		N/A
Product	10,000	10,000	50,000	50% per year	N/A
Customer	5,000	5,000	50,000	100% per year	N/A
Order	0	20,000	500,000	25% per month	7 Years
Order Item	0	100,000	2000,000	25% per month	7 Years
Weekly Sales	0	100	50,000	100% per week	7 Years
Monthly Sales	0	100	10,000	100% per month	7 Years

2.1.3. Business Requirements and Database Design

As mentioned, database design carried out concurrently with system design that designs process of application. In this step of logical database design, we need to consider the database structure with indexes based on business requirements. Both estimated data volume of the table and sample business requirements from system design as below, we need to identify which column may require indexes.

Table 12 - Sample Business Requirement and Logical Database Design

Requirement	Description
Business Process	Sales Order Search
Application Nature	Online Database Search with searching criteria such as date
User Operations	Web GUI interface to search with the conditions specified
System Response Time	Turn around time must be less than 5 seconds
Data Processing Type	Online Transaction Processing (OLTP)
Data Processing Timing	Real-time Data Processing
Concurrency of transaction	Maximum 50 users in peak hours. Average 10 users.
Processing data volume per transaction	Average 100 records
User access control	Management, Sales Department, Accounting Department

2.1.4. Database Optimization

Based on the logical database design process 2.1.1 to 2.1.3, following tasks should be carried out in order to optimize database structure:

- Define indexes
- Add derived data item
- Conduct de-normalization

As for details of indexes are discussed in “Database Programming”. Derived data item, for example, attribute “Age” can be derived from “Date of Birth” may need to be added in the logical database design for the sake of better performance of the database. De-normalization is the process of eliminating normalization as trading off the ideal design in the conceptual database design stage and realistic situation of the database system to be designed in the logical database design.

2.1.5. View

View is a logical schema that filters physical data in the database to end-user should be designed in this stage of database design. Benefits of using view are simplify SQL statements for development team to program and set access privileges to maintain security level for user access to the application systems.

2.1.6. Access Privilege Control

Access privilege control is a key to maintain database security and should be defined in this stage of database design. As for details of access privilege control with MySQL are described in chapter 3. Database Administration with MySQL.

2.1.7. Logical Database Design (ER diagram) mapping to RDBMS

Final stage of the logical database design is to map the ER diagram which was developed during the logical database design process to physical database model. This conversion nowadays executed by ER Modeling tools such as ERWin to generate a script that consists of set of database schema creation command SQL such as CREATE TABLE, CREATE INDEX, CREATE VIEW and so on.

2.1.8. Natural Key and Surrogate Key Models

So far we came down this data modeling or database design process based on Data Oriented

Approach (DOA). However if your application design and development will be carried out with Object Oriented Design and Development Approach, then their could be a mismatch of the data modeling and system application design. One of the solutions that can be considered at the end of the logical database design process, to convert natural primary key for each entity to surrogate key generated by auto increment feature provided by RDBMS. Physical Primary Key of the entity is a running number generated by RDBMS with the auto increment feature and logical primary key in business sense is guaranteed by UNIQUE constraint. Application object such as Java Beans can be mapped with database entity with physical primary key and application looks up logical primary key that is guaranteed by UNIQUE constraint to process business requirement. Figure below depicts sample object model (UML Class Diagram) maps to two approaches such as Natural Key Model and Surrogate Key Model:

Table 13 - Natural Key and Surrogate Key Models

Model	Advantage	Description
Natural Key Model	Seamless to Logical Data Model	Primary Key of the tale is visible to end-user. Ex. Customer ID
Surrogate Key Model	Seamless to Object Oriented Model	Primary Key of the tale is not visible to end-user. Ex. id

2.1.9. Business Rules and Constraint

Based on business rules required to implement from the conceptual data model and business workflow, following constraints in RDBMS need to be reflected in the logical database design:

Table 14 - Constraint and RDBMS

Constraint	Description
NOT NULL	Attribute does not allowed to have NULL value
UNIQUE KEY	Attribute must be unique. Null allowed.
PRIMARY KEY	UNIQUE KEY + NOT NULL
FOREIGN KEY	Foreign key referential constraint to primary key

Same as above, the action rules between entities when event happens need to be reflected in the logical database design as follows:

Table 15 - Action Rules

Action	Description
SET NULL	The Row in dependent entity sets null when the row in parent entity deleted.

RESTRICT	The Row in dependent does not allowed to delete when the row in parent entity deleted.
CASCADE	The Row in dependent entity deleted when the row in parent entity deleted.

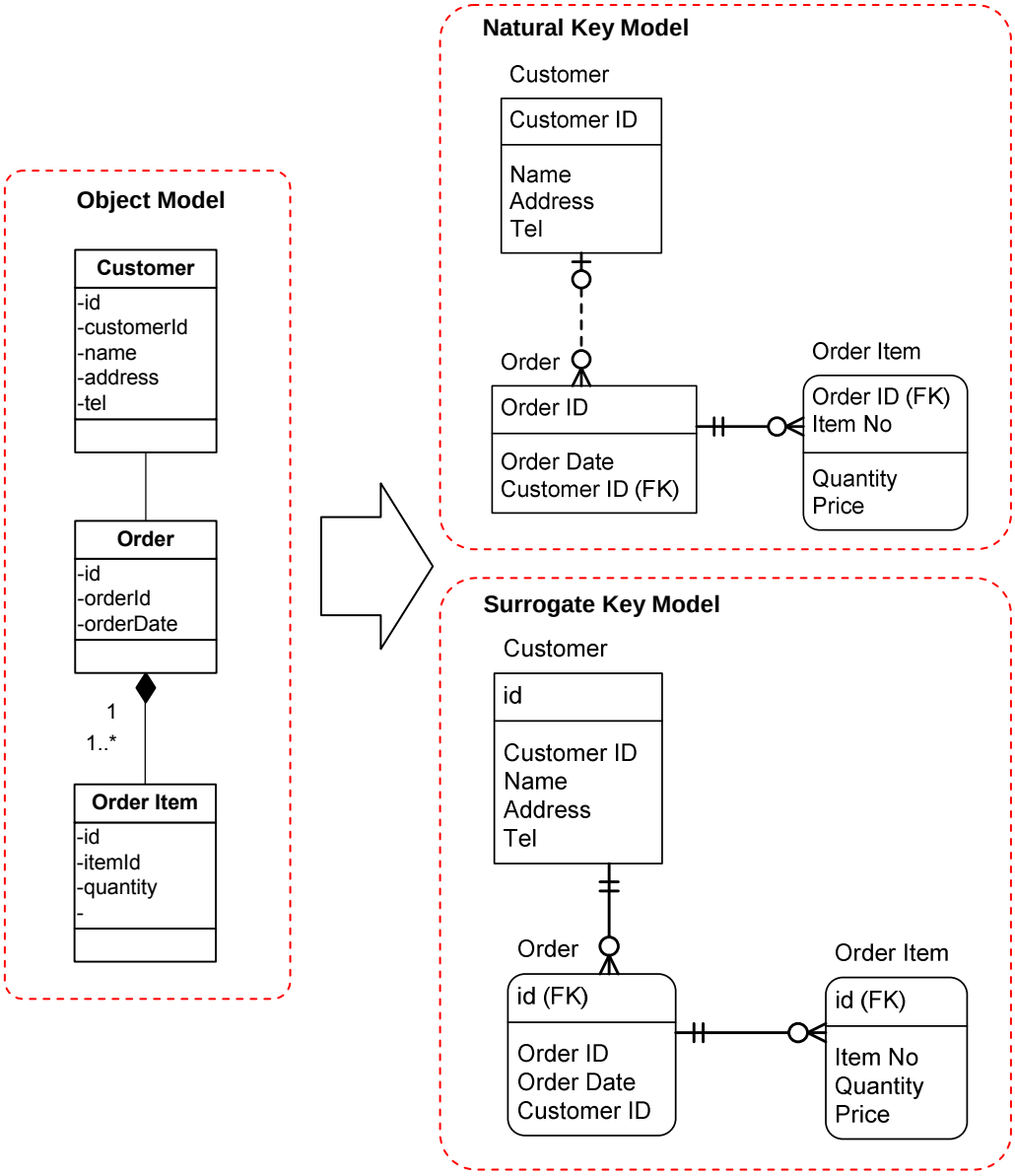


Figure 40 - Natural Key Model and Surrogate Key Model

2.1.10. Logical Database Design Document

List of logical database design document need to be prepared during the design stage is as follows:

Table 16 - List of logical database design document

Document	Description
Logical ER diagrams	Logical ER diagrams
DFD	DFD maps with the logical ER diagrams
CRAD matrix	CRAD matrix maps with the logical ER diagrams
Capacity Planning	Capacity Planning documents
View design	View design documents
Access Privileges Design	Access Privileges Design documents

Exercise 2 - Logical Database Design practice with a case study

1. Conduct logical database design based on the conceptual database design in Exercise 1

- Estimate data volume with capacity planning table
- Design views (Shopping Cart, Order, Stock, Purchase Order, Dispatch Advise, Invoice)
- Design database access privileges
- Finalize the logical database design with ER diagram for MySQL mapping

2.2. Physical Database Design

2.2.1. Introduction to Physical Database Design

Physical database design is the process of selecting disk storage and decide characteristics of data access method to the database. Thus objective of the physical database design is to determine how to utilize hardware resources in order to full fill the user's requirement by implementing reliable database system. In order to design database physically with acceptable performance and reliabilities, you need to take at whole system architecture overview:

- Database server specifications: Clustering, Replication, CPU, Memory and others
- Data storage specifications: Capacity, Seed, Size, RAID configuration and others
- Deployment of application components: Java, Script, Stored Procedure and others
- Deployment of middleware: JDBC driver, ODBC driver, Native driver and others
- Transaction monitoring: TP monitor, EJB monitor and others
- Network bandwidth: Estimation of peak time, non-peak time and others
- Firewall: Access control policy and others
- User authentication: Directory server, CA and others

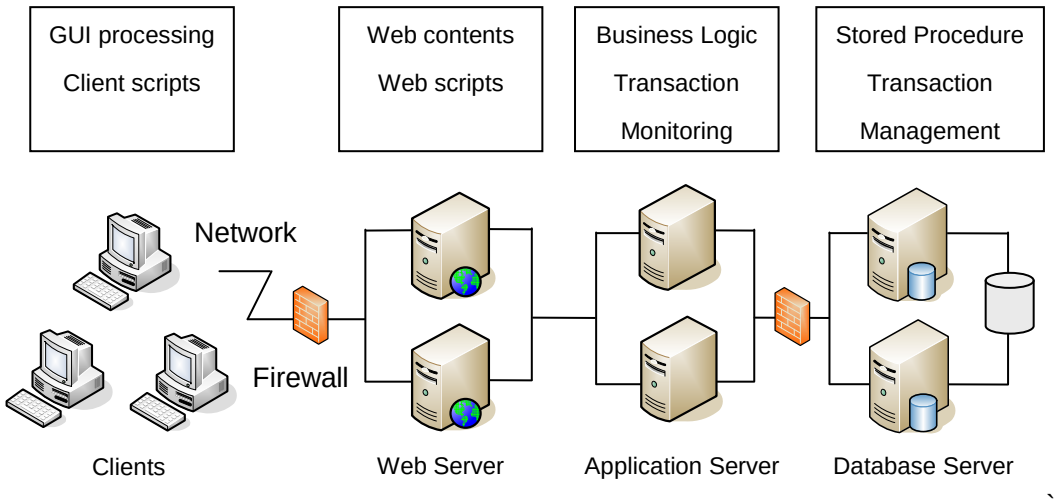


Figure 41 - System architecture overview and database

2.2.2. RDBMS Specification

It is important to review system or application nature when you start the physical database

design. From database management systems point of view, there are types such as Online Transaction Processing (OLTP), DSS (Decision Support System) or OLAP (Online Analytical Processing) as shown in table below:

Table 17 - Type of database and application nature

Criteria	OLTP	DSS (OLAP)
Transaction	Short and many	Long and few
SQL	Heavy DML (Insert, Delete, Update)	Heavy Query (SELECT)
Data Volume	Filtered Data from large volume	Huge volume for accurate analysis

Based on understanding of above database type, assume that MySQL is the database server to be chosen,

- Memory and cache allocation for MySQL server
- File size allocation such as data files, archive (REDO or binary) log files and event/error log files
- Table space and data files

Details of how to configure MySQL server is discussed in Database Administration with MySQL.

2.2.3. Security Consideration

This is one of the most critical part of physical database design processes because the risks are critically high in case if there is a pitfalls of security in the system. Following is a items that you need to take into account to design security:

- Database server OS security
- Network security
- MySQL server security
- Attacking such as SQL injection

2.2.4. Performance Tuning

This is also one of the most critical part of physical database design processes. The performance of the application and database are sometimes (or most of the time) not as good as what you expected based on the estimation of data volume you designed during the logical database design process. Below is some activities need to be conducted in order to catch up the proper performance that should be performed in production system:

- Test with production volume data (maximum) to analyze and optimize SQL statements in allocation with the performance tuning and optimization tools
- Conduct technical sessions about SQL performance up tips to the application developers

2.2.5. Disk Array

Design and configuration of disk array is depend on system's availability and reliability requirement. (Hardware budget allocated is also a factor need to be considered) Table below is a list of the RAID configurations that commonly used today and as long as both reliability and performance are in considerations:

Table 18 - RAID configuration types

RAID type	Description
RAID-0	Striping. It provides fast access time and efficient usage of storage however there is no redundant storage of data.
RAID-1	Mirroring. It provides redundant storage of data however due to parallel writing to multi-disks performance imprecation need to be considered.
RAID-0+1	Striping + Mirroring. It is robust (both performance and redundancy) solution however it could be expensive solution to implement.
RAID-5	Rotating Parity Array or Parity-Striping. It is one of the most popular configurations.

2.2.6. Data Backup and Disaster Recovery

Disaster Recovery (DR) is the process of recovering production system with accessing database, hardware and software necessary to resume critical business operations after a natural or human-caused disaster. Disaster recovery site which is geographically locate remotely away form production site to distribute risks from natural or human-caused disaster. Disaster Recovery Plan (DRP) should be distributed for system administration staff and trained to prepare with those disasters and unexpected or sudden loss of key personnel the focus of which is data protection.

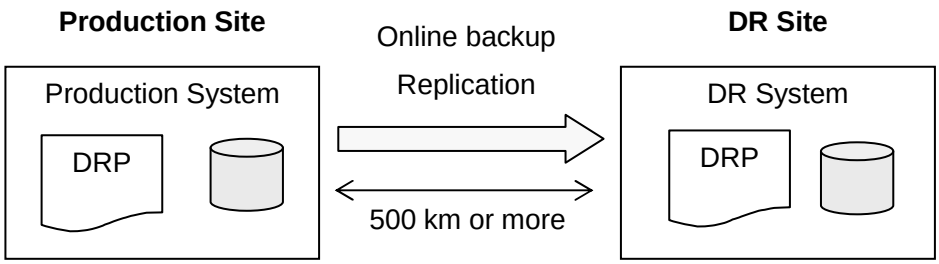


Figure 42 - Disaster Recovery site

2.2.7. Physical Database Design Documents

List of sample physical database design documents need to be prepared during the physical database design stage is as follows:

Table 19 - List of physical database design document

Document	Description
Physical ER diagrams	Physical ER diagrams with DDL for MySQL
Database table space design	Logical databases and physical disk storages mappings
File system design	Size and location of Data file, archive (Binary) log files and event/error log files for each hard disk partitions
Database security design	Server, Network and Database objects access privileges design
MySQL configuration design	MySQL option parameter configuration for my.cnf file
Performance (stress) test plan	Performance (stress) test planning document with production data
Disaster recovery plan	Disaster recovery planning with system administrator's training

Exercise 3 - Physical Database Design practice with a case study

1. Conduct physical database design based on the logical database design in Exercise 2 to come up with physical database design document.

- Draw database servers architecture overview
- Write database server hardware specifications
- Write file system specifications
- Write MySQL server option parameter configuration

3. Database Administration with MySQL

3.1. Introduction to MySQL Architecture for Administration

3.1.1. MySQL Architecture Overview

Understanding of the architecture of RBBMS, MySQL in this case, is the first step of learning how to administrate the database. It is important to have an idea how does RDBMS looks like in both logically and physically, as well as how does it works are the keys for better understanding in order to administrate MySQL databases properly.

First of all, as can be seen figure below, MySQL server logically consists of two levels such as Database management (Upper layer) level and Storage management (Lower layer) level. Queries sent by MySQL clients will be received and processed by Query Processor and Optimizer. Transaction engine and Recovery engine are responsible to handle database transaction and the data will be passed to Storage engine as lower level of MySQL. MySQL storage engine is independent from actual data store. It is called as MySQL pluggable storage engine architecture:

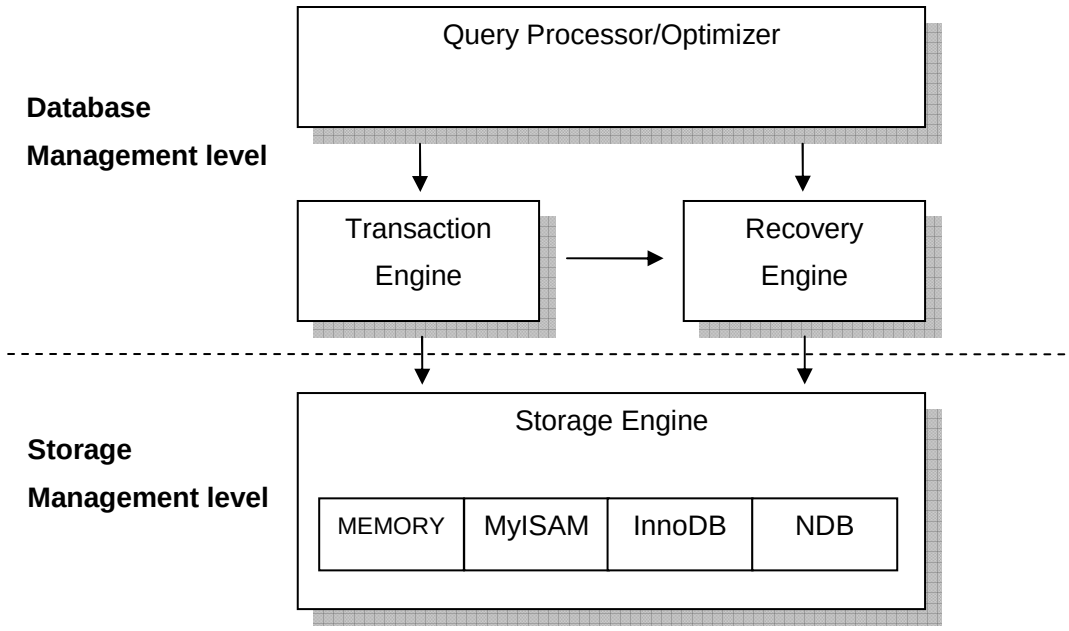


Figure 43 - MySQL Architecture Overview

3.1.2. MySQL Storage Engine

As for physical architecture of MySQL, as mentioned earlier, one of the most unique features is for their storage engine called as pluggable storage engine. In this architecture, type of database table can be chosen and figure below depicts some of commonly used table types and actual storages such as their memory or files.

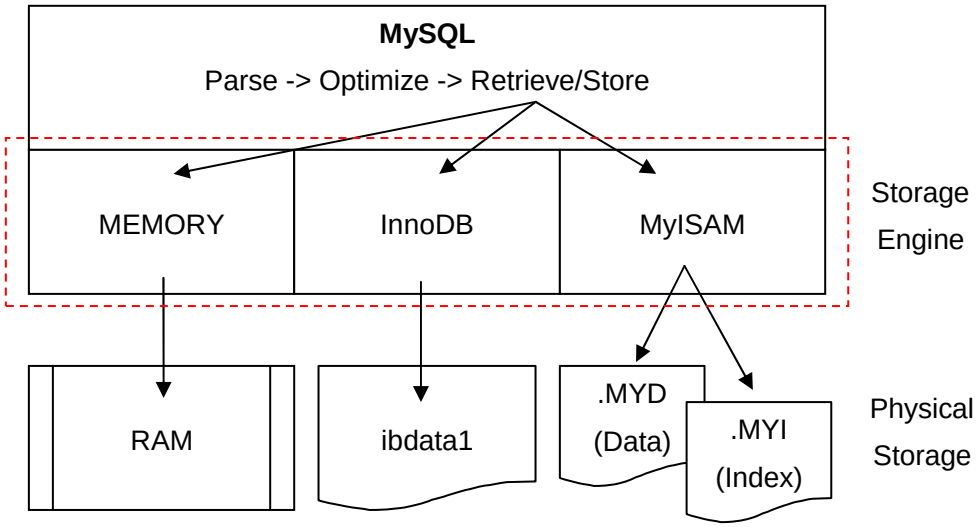


Figure 44 - MySQL Storage Engine

Table below highlights features of those table types discusses above. MEMORY is usually used for temporary purpose databases, MyISAM is for read-only or reference based databases because of the faster access to the database. InnoDB is used for transactional databases that supports ACID with four levels of transaction isolation levels.

Table 20 - MySQL 5.0 Storage Engine

Storage Engine	Transaction	Foreign Key	Full-Text Search	Description
MEMORY	No	No	No	Both data and index are stored in memory (RAM) hence those data and index will be lost when MySQL server shutdown.
MyISAM	No	No	Yes	No transaction supported however it is known as very fast accessing speed will be provided under appropriate configuration. One MyISAM table consists of 3 files such as .frm (Definition), .MYD (Data) and .MYI (Index).
InnoDB	Yes	Yes	No	Four Transaction isolation levels defined by ISO supported. It is known as most functional storage engines as other RDBMS. Table space and REDO log are required for the transaction.

Below is sample CREATE TABLE statements with how to specify MySQL table types using ENGINE option:

```
mysql> CREATE TABLE t1 (id int) ENGINE = MEMORY;
mysql> CREATE TABLE t2 (id int) ENGINE = MyISAM;
mysql> CREATE TABLE t3 (id int) ENGINE = InnoDB;
```

If MySQL tables are created without specifying ENGINE, then table type follows depend on following option stated in my.cnf configuration file:

```
[mysqld]
default-table-type=InnoDB

Note: MySQL table will be created by InnoDB table type in case ENGINE
option was not specified in CREATE TABLE statement.
```

Another example of how to configure default table type in my.cnf configuration file. If this variable is to enable InnoDB storage engine to creates each new table using its own .ibd file for storing data and indexes instead of creating it in the shared table space:

```
[mysqld]
innodb_file_per_table
```

3.1.3. MySQL Directory Structure

Second stop of understanding MySQL could be looking at directory structure of MySQL. Following table shows list of sub directory under /usr/local/mysql with non RPM package installation to install MySQL 5.0 to /usr/local/ folder:

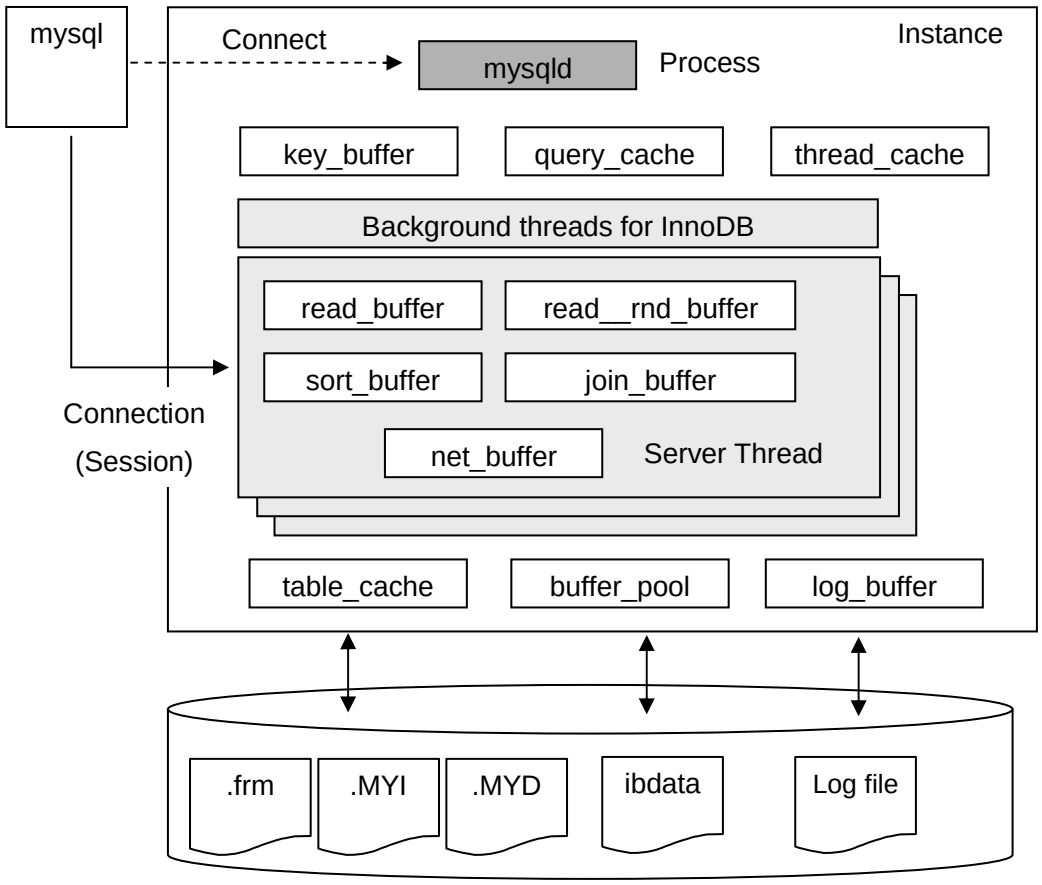
Table 21 - MySQL installation directory by non RPM package

Directory	Description
bin/	Binary commands
data/	Data directory (datadir)
include/	Header files
lib/	Library files
scripts/	Script files
share/	Fill_help_tables.sql, mysql_fix_privilege_table_sql
share/mysql/	Error message files
support-files/	Mysql.server, my.cnf
docs/	Documents
man/	Online Manual
sql-bench/	Benchmark test
mysql-test/	Mysql-test script

tests/	Test scripts

3.1.4. MySQL Instance Architecture (Process, Thread and Memory)

Understanding of MySQL Server instance is also important for database administration. MySQL instance generally here means that it consists of both MySQL process, threads and MySQL memories. The figure below depicts MySQL instance architecture which consists of, as mentioned, process, threads and memories. As can be seen, MySQL supports multi-threads architecture. For example, once mysqld (MySQL demon process) receives a connection request from mysql (MySQL client process), mysqld creates a thread which consists of several buffer memories. It means that the thread will be created based on each client request under the control of mysqld. That is how MySQL implements multi-user, multithreads by each session.



Note: .frm: Table definition, .MYI: MyISAM index, .MYD: MyISAM data, ibdata: InnoDB

Figure 45 - MySQL Memory Structure

Table below is the list of MySQL buffers and cache memories described the figure above. Those

buffers and memories are configurable with MySQL configuration file (my.cnf).

Table 22 - MySQL buffer and cache memory

Buffer	Description
key_buffer	The size for buffer used for index block. Increase to get better performance as possible as server can allocate physical memory. This buffer is shared by all threads.
read_buffer	The size of buffer by thread used for sequential scan allocation.
read_rnd_buffer	The size of buffer by thread used for random scan allocation.
sort_buffer	The size of buffer by thread used for sorting data. Mainly used for ORDER BY.
join_buffer	The size of buffer by thread used for table joins that do not use indexes and thus perform full table scans.
net_buffer	The size of buffer by thread used for client connection.
query_cache	The size of memory allocated for caching query results by SELECT statement. The default value is 0, which disables the query cache.
thread_cache	The size of memory allocated for caching a thread when a client disconnected. The default value is 0, which disables the query cache.
table_cache	The size of memory allocated for caching opened tables. table_cache, max_connections, and max_tmp_tables system variables affect the maximum number of files the server keeps open
buffer_pool	The size of memory allocated for buffering InnoDB to use to cache data and indexes of its tables.
log_buffer	The size of memory allocated for buffering InnoDB transaction logs.

Please refer to MySQL Reference Manual (Option and Variable Reference) for more detail.

3.2. MySQL Server Management

3.2.1. Installation Methods

There are basically three types of installations are available such as RPM package, Binary (non RPM) package and source code installations. Advantages and disadvantages for each installation method are stated as following table:

Table 23 - MySQL Installation

Installation	Advantages	Disadvantages
RPM package	Easy installation. Creation of user and initialization of database will be automatically carried out by rpm.	No flexibility. Installation directory is fixed and only one server can be installed.

Binary package	No compilation required and Installation directory is configurable.	User creation and database initialization required manually.
Source code package	Flexibility of installation is provided.	Required C/C++ compiler and it take longer time and effort is needed for source code compilation.

Production database servers are normally installed by source code is because best performance must be required with appropriate variables and parameter configurations.

3.2.2. Management Tools

- mysqld

mysqld (mysql demon) is MySQL main process. It can be started using mysqld directly however a wrapper script of mysqld called mysqld_safe is provided to start mysqld safely as following syntax:

```
localhost# mysqld_safe -user=mysql &


localhost# ps -em | grep mysql
 4331 ?    00:00:00 mysqld_safe
 4380 ?    00:00:00 mysqld
```

- mysql

mysql is probably most commonly used tool when you need to administrate MySQL server. It is highly recommended to be familiar with this tool. Please read MySQL Reference Manual for more details:

```
linux>mysql [Option] [Database]
```

Table 24 - mysql options

Option	Description
--user=*User name* -u *User name*	Specify MySQL user name
--password [=*Password*] -p[*Password*]	Specify password for the user. For the security reason, it is recommended to enter password separately.
--host=*Hostname* -h *Hostname*	Specify MySQL Server hostname or IP address
--port=*Port Number* -P *Port Number*	Specify MySQL Server port number. Default port number would be 3306

Example of connecting through TCP port 3306 in localhost with root user using mysql:

```
linux> mysql -u root -p -h localhost -P 3306
```

mysql as a Linux command, it can be used with redirect and pipe features. Table below is a list of sample usage such as loading text file and writing SQL result set to text file:

Table 25 - mysql command with redirect and pipe

Command	Description
lunux> mysql -u root -p test < sql.txt	Load sql.txt to mysql command
lunux> mysql -u root -p test > result.txt	Overwrite mysql command result to result.txt
lunux> mysql -u root -p test >> result.txt	Append mysql command result to result.txt
lunux> echo "SELECT …" | mysql -u root -p test	Send echo to mysql command

Typing '?' displays list of command help in mysql as follows. That is first step to be familiar with mysql command:

```
mysql> ?

List of all MySQL commands:
Note that all text commands must be first on line and end with ';'
?    (\?) Synonym for `help'.
clear  (\c) Clear command.
connect (\r) Reconnect to the server. Optional arguments are db and host.
delimiter (\d) Set statement delimiter.
ego   (\G) Send command to mysql server, display result vertically.
exit   (\q) Exit mysql. Same as quit.
go    (\g) Send command to mysql server.
help   (\h) Display this help.
notee  (\t) Don't write into outfile.
print  (\p) Print current command.
prompt  (\R) Change your mysql prompt.
quit   (\q) Quit mysql.
rehash  (\#) Rebuild completion hash.
source  (\.) Execute an SQL script file. Takes a file name as an argument.
status  (\s) Get status information from the server.
tee   (\T) Set outfile [to_outfile]. Append everything into given outfile.
use   (\u) Use another database. Takes database name as argument.
charset (\C) Switch to another charset.
warnings (\W) Show warnings after every statement.
nowarning (\w) Don't show warnings after every statement.
```

Typing '?' with SQL statement such as SELECT, CREATE displays details of those statement as online help as follows:

```
mysql> ? SELECT
```

```
Name: 'SELECT'
Description:
Syntax:
SELECT
  [ALL | DISTINCT | DISTINCTROW ]
   [HIGH_PRIORITY]
   [STRAIGHT_JOIN]
   [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
   [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
  select_expr, ...
  [FROM table_references
  [WHERE where_condition]
  [GROUP BY {col_name | expr | position}
   [ASC | DESC], ... [WITH ROLLUP]]
  [HAVING where_condition]
  [ORDER BY {col_name | expr | position}
   [ASC | DESC], ...]
  [LIMIT {[offset,] row_count | row_count OFFSET offset}]
  [PROCEDURE procedure_name(argument_list)]
  [INTO OUTFILE 'file_name' export_options
   | INTO DUMPFILE 'file_name'
   | INTO @var_name [, @var_name]]
  [FOR UPDATE | LOCK IN SHARE MODE]]

SELECT is used to retrieve rows selected from one or more tables,
and can include UNION statements and subqueries.
```

SHOW is a command to list down specified database object. As can be seen, together with DATABASES or TABLES displays list of databases and tables:

```
mysql> SHOW DATABASES;
mysql> SHOW TABLES;
mysql> SHOW COLUMNS FROM table;
```

USE is a command to select database as follows:

```
mysql> USE test;
Database changed
```

- mysqladmin

mysqladmin is a main command to administrate MySQL server with following syntax:

```
linux>mysqladmin [Option] command
```

Table below is a list of command can be used with mysqladmin:

Table 26 - mysqladmin commands

Command	SQL	Description
create	CREATE DATABASE	Create a new database
drop	DROP DATABASE	Delete existing database
password	SET PASSWORD	Change password for user
ping		Check MySQL server process (alive or dead)
processlist	SHOW PROCESSLIST	Display list of MySQL server threads
status	SHOW STATUS	Display status of MySQL server
variable	SHOW VARIABLES	Display list of MySQL server variables
version	SELECT VERSION()	Display MySQL server version
shutdown		Shutdown MySQL server
kill	KILL	Kill MySQL server
reload		Reload privileges tables
refresh		Close all table and log files and reopen it

Examples of viewing MySQL server details using mysqladmin commands:

```
linux> mysqladmin -u root -p ping
linux> mysqladmin -u root -p version
linux> mysqladmin -u root -p variables
linux> mysqladmin -u root -p processlist
linux> mysqladmin -u root -p status
```

Examples of administrating MySQL server:

```
-- Start MySQL server
linux# /usr/local/bin/mysqld_safe -user=mysql &
linux> mysqladmin -u root -p reload
linux> mysqladmin -u root -p refresh
-- Stop MySQL server
linux> mysqladmin -u root -p shutdown
linux> mysqladmin -u root -p kill 20
-- Start MySQL server again
linux# /usr/local/bin/mysald_safe -user=mysql &

Note: 20 is Process ID which can be found from Process List.
```

- mysqldump

mysqldump is command to backup database. Please refer to Database Backup and Restore for more detail.

- mysqlbinlog

mysqlbinlog is a command to view MySQL binary log files. Please refer to MySQL Logs for more detail.

- GUI tools

Useful GUI tools are provided by MySQL AB to support MySQL database design, development, administration and database migration:

Table 27 - MySQL GUI Tools

GUI Tool	Description
MySQL Administrator	MySQL server administration tool like pgAdmin III for PostgreSQL.
MySQL Query Browser	MySQL client Query tool like pgAdmin III Query Tool for PostgreSQL.
MySQL Workbench	MySQL database design tool with E-R Modeling such as ERWin, Microsoft Visio.
MySQL Migration Toolkit	MySQL server migration tool to migrate from other databases to MySQL database.

MySQL Administrator is one of those GUI tools that can administrate MySQL server. Following screen captures are some of examples that how MySQL Administrator manages MySQL:

Figure 46 - Logon to MySQL Administrator

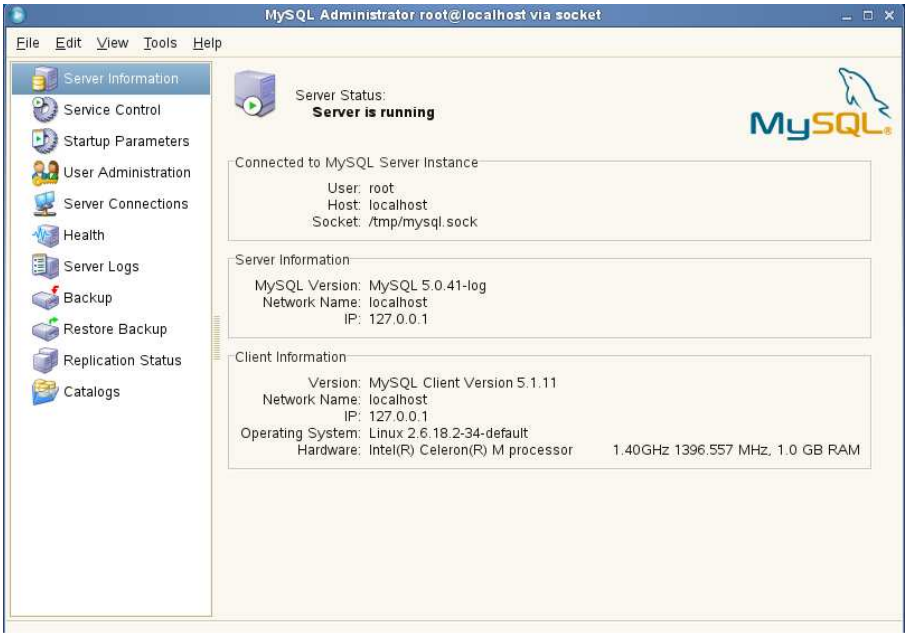


Figure 47 - MySQL Administrator user interface

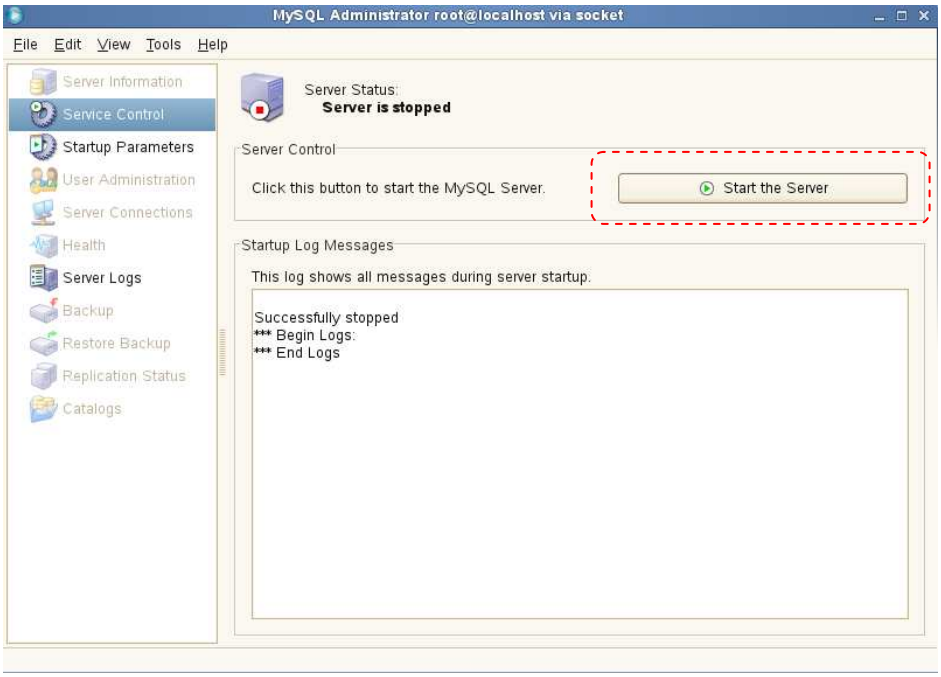


Figure 48 - Start MySQL Server

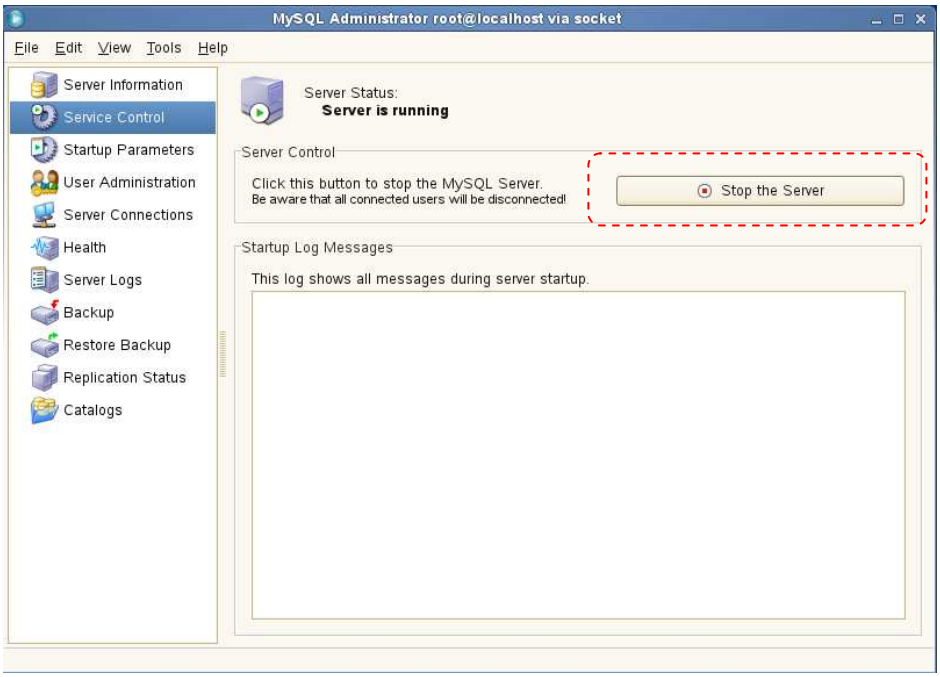


Figure 49 - Stop MySQL Server

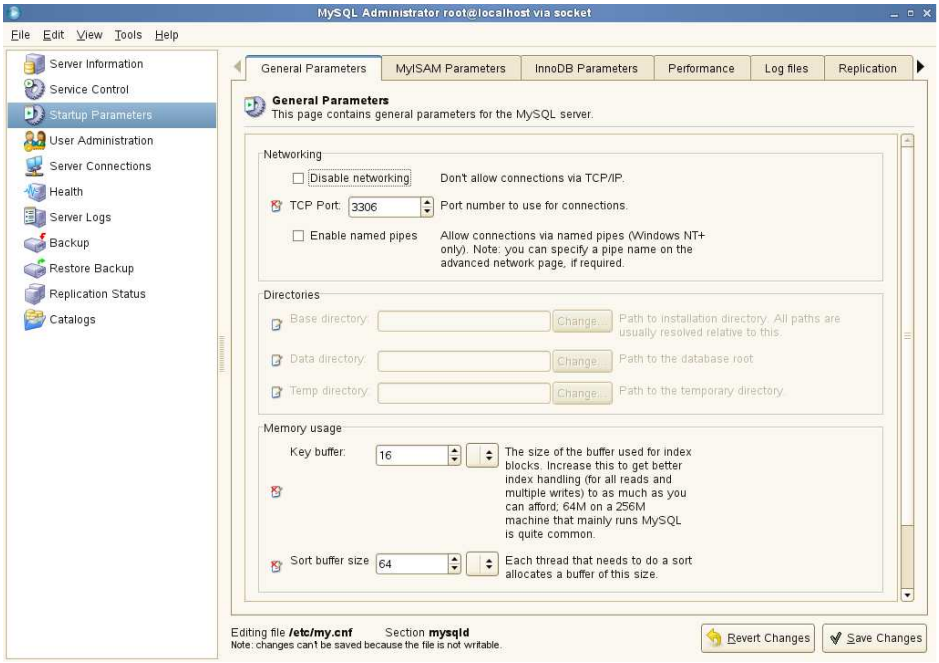


Figure 50 - MySQL Startup (Option) parameters setting

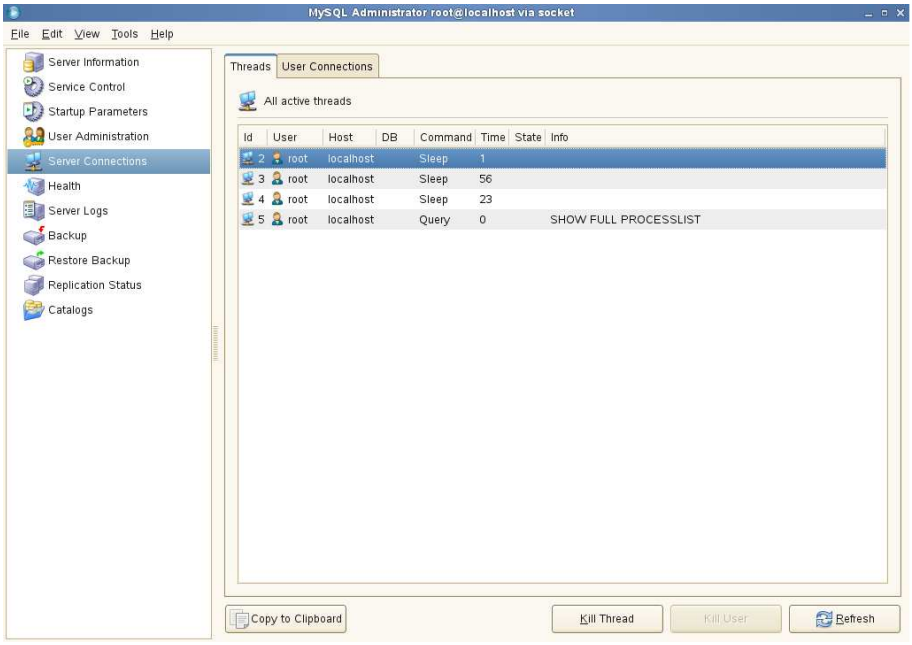


Figure 51 - MySQL Threads and Users monitoring

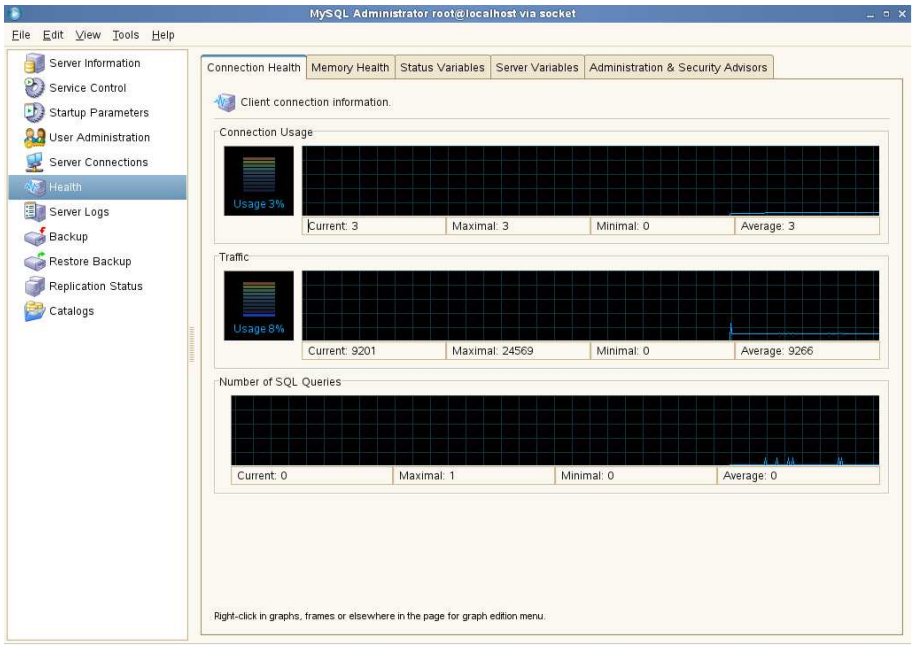


Figure 52 - MySQL Server health checking

3.3. MySQL User Management

3.3.1. Security Management

Not only MySQL but even all the database must be protected from access by unauthorized users. MySQL as same as most of other RDBMS provides following security features:

- Password authentication

  MySQL encrypts passwords using its own algorithm. There are several ways to set the password for MySQL user such as:

  - GRANT … IDENTIFIED BY
  - SET PASSWORD
  - UPDATE user table
  - mysqladmin

- Access rights control

  It is discussed in Access Privilege Control.

- Audit trails

  MySQL General Log can capture all user activities for audit purposes.

- Data encription

  MySQL5.0 non-RPM (Binary) package supports SSL encryption feature to be configured. Please refer to MySQL Reference Manual for details

3.3.2. Access Privilege Control

MySQL has own users to control access privileges to database objects. Those users are different from Linux users. MySQL users are also associated with host that runs MySQL server. For example, root user in localhost describes as root@localhost. In other words, if tow MySQL servers has both ‘scott’ user however those two ‘scott’ are treated as different users. Table below shows MySQL users and database access privileges:

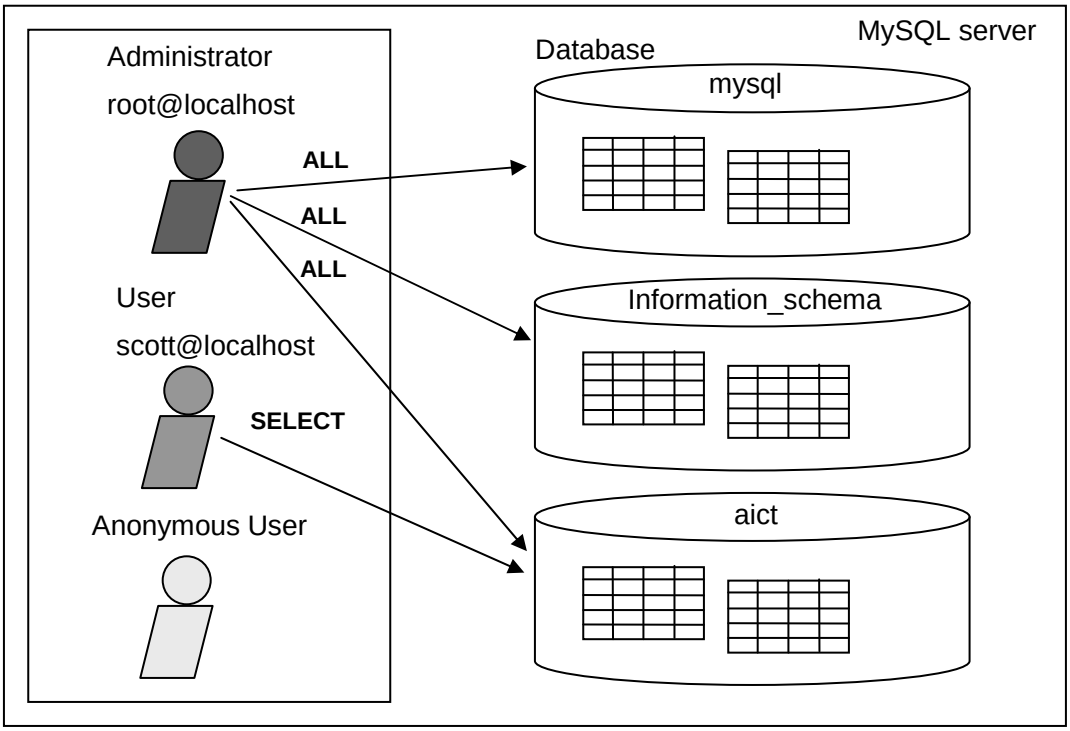


Figure 53 - MySQL server, database, user and privileges

MySQL users are maintained by user table in mysql database:

```
mysql> SELECT user, host FROM mysql.user;
+-------+-----------+
| user | host   |
+-------+-----------+
|    | localhost | <-- Anonymous user
| root | localhost |
| scott | localhost |
+-------+-----------+
3 rows in set (0.05 sec)
```

Table below is a list of tables in mysql database that maintains access privilege control:

Table 28 - MySQL Access Privilege control tables in mysql database

Table	Description
user	User definition table
host	Host privileges definition table
db	Database privileges definition table
tables_priv	Table privileges definition table

columns_priv	Columns privileges definition table

Table 29 - MySQL Access Privileges

Privilege	Description
[General]	
ALTER	Alter table and field
CREATE	Create database and table
CREATE TEMPORARY TABLES	Create temporary table
DELETE	DELETE statement
DROP	Delete database and table
FILE	LOAD DATA and SELECT … INTO OUTFILE statements
INDEX	Create and drop index
INSERT	INSERT statement
LOCK TABLES	Table lock
SELECT	SELECT statement
SHOW DATABASES	Display list of database
UPDATE	UPDATE statement
[Stored Routine]	
EXECUTE	Execute stored procedure
CREATE ROUTINE	Create stored routine
ALTER ROUTINE	Alter stored routine
[View]	
CREATE VIEW	Create view
SHOW VIEW	Display list of view
[Privilege]	
ALL PRIVILEGES	All privileges
GRANT	Grant privileges to other users
CREATE USER	Create user
[Database Administrator]	
PROCESS	Display list of process
SUPER	Execute mysqladmin kill command
RELOAD	Reload and flush command
SHUTDOWN	Execute mysqladmin shutdown command
REPLICATION CLIENT	Replication client
REPLICATION SLAVE	Replication slave
[Others]	
USAGE	No privileges
REFERENCES	Table reference

3.3.3. User Management

In order to manage MySQL users, you need to logon with root@localhost user account to MySQL server. Below is a sample SQL to manage user account:

Create a user:

```
mysql> CREATE USER scott@localhost IDENTIFIED BY 'tiger';
```

Grant SELECT privilege to the user to access ictti database:

```
mysql> GRANT SELECT ON aict.* TO scott@localhost
        IDENTIFIED BY 'tiger'
```

Display privileges given to the user:

```
mysql> SHOW GRANTS FOR scott@localhost;

mysql> SELECT * FROM information_schema.user_privileges
        WHERE grantee LIKE '''scott%';
```

Remove privileges from the user:

```
mysql> REVOKE SELECT ON aict.* FROM scott@localhost;
```

3.4. MySQL Database Backup and Restore

Database backup and restore are important technique not only for production database administration but also necessary to carry out during software development and test phases. There are several ways of MySQL database backup and restore are available such as:

Table 30 - MySQL Database Backup and restore Methods

Methods	Description
Cold Backup and restore	Database file backup and restore using Linux commands such as cp, tar. Database must be shutdown before taking backup and restore.
Online Backup and restore	Database backup using MySQL tools such as mysqldump or using SQL statements such as SELECT ,,, INTO OUTFILE. Database restore using mysql or SQL statement such as LOAD DATA INFILE.
Others	Database backup and restore using MySQL database replication features to replicate slave database in remote site.

3.4.1. Cold backup

Cold backup is usually to use Linux commands such as tar, cp to copy datadir (/usr/local/mysql/data) to backup location after shutting down MySQL server. Below is an example cold backup using cp command:

```
linux> mysqladmin -u root -p shutdown
linux> cp -rp /usr/local/mysql/data /backup/mysql-backup
linux# /usr/local/mysql/bin/mysqld_safe --user=mysql

Note: MySQL server need to shutdown before backup. The server needs
to be restarted after the backup.

cp command options:
-r: Copy directories recursively.
-p: Preserve all information, including owner, group, permissions,
and timestamps.
```

3.4.2. mysqldump

mysqldump is a backup tool to backup MySQL database(s) or specific table(s). The backup file consists of DML statements such as CREATE TABLE and INSERT to allow restore to the original tables and databases.

Syntax:

```
linux> mysqldump [Option] database [table …]
linux> mysqldump [Option] --databases [Option] database [database …]
linux> mysqldump [Option] -all-databases [Option]
```

Examples:

- Back up mytable table in mydatabase database

```
linux> mysqldump -u root -p mydatabase mytable > backup.txt
```

- Restore mytable in mydatabase database from back up text file

```
linux> mysql -u root -p mydatabase < backup.txt
```

Please refer to MySQL Reference Manual for detail options.

3.4.3. SELECT … INTO OUTFILE / LOAD DATA INFILE

SELECT … INTO OUTFILE is the statement to write a table to a file as syntax and example stated below:

Syntax:

```
SELECT field [, field …] INTO OUTFILE 'file name'
```

Example:

```
-- All the rows and columns in mytable will be written in mytable.txt
mysql> USE mydatabase;
mysql> SELECT * FROM mytable INTO OUTFILE ‘mytable.txt’

linux> ls -la /usr/local/mysql/data/mydatabase/mytable.txt
```

LOAD DATA INFILE is the complement statement of SELECT … INTO OUTFILE to read specified text file and insert it into a table specified as stated below:

Syntax:

```
LOAD DATA INFILE 'file name' INTO TABLE table name
```

Example:

```
-- All the data in mytable.txt will be loaded into mytable
mysql> USE mydatabase;
mysql> LOAD DATA INFILE ‘mytable.txt’ INTO TABLE mytable

mysql> SELECT * FROM mytable;
```

3.5. MySQL Configuration File

MySQL tools such as mysqld, mysql, mysqladmin and others can read their startup options from a configuration file that can be used as a basis for tuning and optimizing MySQL server. The configuration file provides a convenient way to specify commonly used options so that it is not necessary to set with command line each time it is executed. The configuration file in Linux environment usually called “my.cnf”.

3.5.1. File Format

Table below is basic format of the configuration file (my.cnf):

Table 31 - MySQL Configuration File (my.cnf) Format

Format	Description
#	Comment
[Group name]	Group declaration such as [mysqld], [mysql], [mysqldump], [client] and [mysqld_safe]
Option	Same as - - option from command line. Ex. flush_logs
Option = Value	Same as - - option = value from command line. Ex. open-file-limit = 4096

3.5.2. File Location

The configuration file (my.cnf) can be found following server locations. The sequence number is priority of loading by MySQL server and last loaded file overwrites previous configuration setting:

Table 32 - my.cnf location

No	Location	Description
1	/etc/my.cnf	Linux /etc
2	$MYSQL_HOME environment variable	Specified by MySQL environment variable
3	-defaults-extra-file =	Specified by mysqld option
4	~/.my.cnf	Linux ~/.

3.5.3. Sample my.cnf File

Below is sample my.cnf file. Please refer to MySQL Reference Manual for details of parameters:

```
# Example MySQL config file for small systems.

# The following options will be passed to all MySQL clients
[client]
#password = your_password
port = 3306
socket = /tmp/mysql.sock
# The MySQL server
[mysqld]
port = 3306
socket = /tmp/mysql.sock
skip-locking
key_buffer = 16K
max_allowed_packet = 1M
table_cache = 4
sort_buffer_size = 64K
read_buffer_size = 256K
read_rnd_buffer_size = 256K
```

```
net_buffer_length = 2K
thread_stack = 64K
server-id   = 1
log-bin=mysql-bin
innodb_data_home_dir = /usr/local/mysql/data/
innodb_data_file_path = ibdata1:10M:autoextend
innodb_log_group_home_dir = /usr/local/mysql/data/
innodb_log_arch_dir = /usr/local/mysql/data/
innodb_buffer_pool_size = 16M
innodb_additional_mem_pool_size = 2M
innodb_log_file_size = 5M
innodb_log_buffer_size = 8M
innodb_flush_log_at_trx_commit = 1
innodb_lock_wait_timeout = 50
innodb_file_per_table
[mysqldump]
quick
max_allowed_packet = 16M
[mysql]
no-auto-rehash
safe-updates
[isamchk]
key_buffer = 8M
sort_buffer_size = 8M
[myisamchk]
key_buffer = 8M
sort_buffer_size = 8M
[mysqlhotcopy]
interactive-timeout
```

3.6. MySQL Database Maintenance

MySQL provides to check, repair, or optimize database tables (MySQL data files).

3.6.1. Table Checking

CHECK TABLE checks table errors and reports if errors are existed as following syntax:

```
mysql> CHECK TABLE table name [Option]
```

3.6.2. Table Repairing

REPAIR TABLE repairs table errors if exists. This is applicable only for MyISAM table type:

```
mysql> REPAIR TABLE table name [Option]
```

3.6.3. Table Optimization

OPTIMAIZE TABLE optimizes table with following syntax:

```
mysql> OPTIMAIZE TABLE table name [Option]
```

Example of Database table maintenance using CHECK, REPAIR and OPTIMIZE TABLE:

```
mysql> LOCK TABLE mytable WRITE;
mysql> FLUSH TABLES;
mysql> CHECK TABLE mytable;
mysql> REPAIR TABLE mytable;
mysql> OPTIMAIZE TABLE mytable;
mysql> UNLOCK TABLES;
```

3.7. MySQL Log Files

3.7.1. MySQL Server Log Types

MySQL provides functions to record down what is happening with mysqld in several ways in different files for logging purposes. All those log files can be created under datadir variable (/usr/local/mysql/data) by default. Table below shows some of those commonly used logs:

Table 33 - MySQL Server Log Types

Log type	mysqld option or my.cnf	Description
Error Log	--log-error [mysqld] log-error	Error Log captures event for start and stop **mysqld** as well as critical errors happened at **mysqld**.
Slow Query Log	--log-slow-queries [mysqld] log-slow-queries	Slow Query Log captures SQL statements that take more than the specified time to execute. The time can be specified at **long_query_time** option.
Detail Query Log	--log [mysqld] log	Detail Query Log captures event to identify what **mysqld** is doing such as client connections and executed queries by each client.
Binary Log	--log-bin [mysqld] Log-bin	Binary Log captures specifically DML (without SELECT) statements. This log is recorded not text file but binary file and it can be used for roll-forward recovery process.

Note: Above logs can be activated either by mysqld options or specified under [mysqld] group in my.cnf files.

3.7.2. Error Log

How to enable Error log is as follows. Default log file name is hostname.err:

```
--log-error [= File name]

[mysqld]
log-error [= File name]
```

3.7.3. Slow Query Log

How to enable and configure query time (Default long-query-time is 10 second) is are follows. Default file name is hostname-slow.log and the file will be created under datadir folder:

```
--log-slow-queries [= File name]
long-query-time = [time(second)]

[mysqld]
log-slow-queries [= File name]
long-query-time = [time(second)]
```

3.7.4. Detail Query Log

How to enable detail query log is as follows. Default file name is hostname.log and the file will be created under datadir folder:

```
--log [ = File name]

[mysqld]
log [= File name]
```

3.7.5. Binary Log

How to configure binary log is as follows. Default file name is hostname-bin.NNNNNN (NNNNNN means six-digit number) and the file will be created under datadir folder::

```
--log-bin [= File name]

[mysqld]
log-bin [= File name]
```

mysqlbinlog is a MySQL command that allow to view the binary log file. Because the binary log file is not text format to view by text editors. How to view the binary log using mysqlbinlog command is as follows:

```
linux>mysqlbinlog -u root -p localhost-bin.000001
```

Table below is a list of options available for mysqlbinlog command:

Table 34 - mysqlbinlog option

Option	Description
start-datetime=	Filter for start date and time for binary log
stop-datetime=	Filter for stop date and time for binary log
start-position=	Filter for start event position for binary log
stop-position=	Filter for stop event position for binary log
-o, --offset=	Offset number of event specified
-d, --database=	Logs specified database

3.7.6. Sample Log Files

Below is sample my.cnf configuration in order to activate above log files:

```
-- Sample my.cnf file
[mysqld]
log
log-error
log-slow-queries
long-query-time=2
log-bin

# Note: Detail, Error, Slow Query and Binary log files will be created
under datadir (/use/local/mysql/data/) directory.
```

Below is sample error log file. This log file is important as well as useful in case if MySQL server encounters some trouble. The sample shows that MySQL server could not start up due to some type errors in my.cnf file:

```
# SERVER SECTION
#
[mysqld]

my_error

# The TCP/IP Port the MySQL Server will listen on
port=3306
innodb_file_per_table
```

```
070501 13:30:11 mysqld started
070501 13:30:12 [ERROR] /usr/local/mysql/bin/mysqld: unknown option
'my_error'
070501 13:30:13 mysqld ended


Note: ERROR was recorded and mysqld was not started because my_error
is not recognize option for MySQL.
```

Below is sample slow query log file. This log file is useful when database performance tuning is required:

```
# Time: 070518 14:53:52
# User@Host: root[root] @ localhost [127.0.0.1]
# Query_time: 26 Lock_time: 0 Rows_sent: 40392000 Rows_examined: 8000
use myengine;
SELECT * FROM t1, t2;
# Time: 070518 14:57:12
# User@Host: root[root] @ localhost [127.0.0.1]
# Query_time: 2 Lock_time: 0 Rows_sent: 680000 Rows_examined: 40
SELECT * FROM t3,t4;
use myengine;

Note: Above Cross join SELECT statements were captured as the slow
queries because it took more than 2 seconds to process.
```

Below is sample detailed log file. As can be seen, each one of database activity was captured as details as possible:

```
070504 10:55:00      1 Connect  root@localhost on
                   1 Query   SELECT @@sql_mode
                   1 Query   SET SESSION
sql_mode='STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION'
                   1 Query   SET NAMES utf8
                   2 Connect  root@localhost on
                   2 Query   SELECT @@sql_mode
                   2 Query   SET SESSION
sql_mode='STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION'
                   2 Query   SET NAMES utf8
                   2 Quit
070504 10:55:11      3 Connect  root@localhost on
                   3 Query   show variables like '%buffer%'
070504 10:55:41      1 Connect  root@localhost on
                   1 Query   SELECT @@sql_mode
                   1 Query   SET SESSION
sql_mode='STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION'
                   1 Query   SET NAMES utf8
070504 10:55:46      2 Connect  root@localhost on
```

```
                  2 Query   show variables like '%buffer%'
070504 11:07:50      2 Query   show variables like '%cache%'
070504 11:25:27      2 Query   show variables like '%buffer'
070504 17:26:36      2 Query   show databases
070504 17:26:41      2 Query   SELECT DATABASE()
070504 17:26:45      2 Query   show tables
070504 17:26:54      2 Query   show create table account
070504 17:27:51      2 Query   show index from account
070504 18:08:11      2 Query   help 'ANALYZE'
070504 18:08:22      2 Query   help 'analyze table'
070505 8:51:26       5 Query   create databse s-sd-b
070505 8:52:24       5 Query   drop database s-sd-b
```

Below is sample binary log file viewed by mysqlbinlog. As can be seen, all the DML statements were captured in this log file:

```
-- Sample Binary Log file
use aict/*!*/;
SET TIMESTAMP=1179958689/*!*/;

DROP TABLE IF EXISTS `book`/*!*/;
# at 3176
#070524 7:18:09 server id 1 end_log_pos 3487       Query
    thread_id=29exec_time=0   error_code=0
SET TIMESTAMP=1179958689/*!*/;

CREATE TABLE `book` (
 `id` int(11) NOT NULL,
 `title` varchar(50) default NULL,
 `isbn` varchar(50) default NULL,
 `price` int(11) default '0',
 `author_id` varchar(50) default NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8/*!*/;
# at 3487
#070524 7:18:09 server id 1 end_log_pos 3586       Query
    thread_id=29exec_time=0   error_code=0
SET TIMESTAMP=1179958689/*!*/;

INSERT INTO `book` VALUES (1,'Introduction to
MySQL','1-1111-1111-1',285,'01'),(2,'Introduction to
PostgreSQL','1-1111-1111-2',300,'01'),(3,'Introduction to
Database','1-1111-1111-3',500,'01'),(4,'Introduction to
Java','1-1111-1111-4',750,'02')/*!*/;
# at 4268
#070524 7:18:10 server id 1 end_log_pos 4295       Xid = 588
COMMIT/*!*/;
# at 4295
```

```
#070524 7:26:05 server id 1 end_log_pos 4430       Query
    thread_id=29exec_time=0   error_code=0
SET TIMESTAMP=1179959165/*!*/;
```

3.8. MySQL Troubleshooting Tips

Followings are MySQL troubleshooting tips and general recommendations to go through when you encountered some technical problems.

3.8.1. Find out Root Cause

First thing you need to do, when problem arise, is to find out which programs, services or environment is/are causing problems. For example, if mysql client can not connect to MySQL server, check the mysqld process is existing or not probably using ps command or mysqladmin before asking your lecturer for help.

3.8.2. Read Error Messages

Read carefully error message given by MySQL. This is one of the most important hint to solve your problem. Error Code are explained in MySQL Reference Manual: Appendix B. Errors, Error Codes, and Common Problems.

3.8.3. Read Log Files

Examine MySQL log files especially for error log file. For example, if MySQL server won't be able to start, the error message should be captured in the error log. Do get into the habit of checking log files each time when problem arise.

3.8.4. Read Reference Manual

MySQL Reference Manual is a kind of Bible for IT people those who are using MySQL for their projects. Please read the manual as possible as you can in order to find out which part of the manual to be helpful when you have problems.

3.8.5. Browse Technical Discussion Forums

There might be some other IT people who are also using MySQL has same experience that you are encountering. Following Web sites are some of those technical dissection forums for MySQL you may found the solution you are looking for:

Table 35 - MySQL Forums

Forum	URL
MySQL Forums	http://forums.mysql.com/
Dev Shed Forums	http://forums.devshed.com/forumdisplay.php?forumid=4
DaniWeb IT Discussion Community	http://www.daniweb.com/techtalkforums/forum126.html

Exercise 4 - MySQL Database Administration practice

1. Architecture

- Storage engine (Table Type)
  1. Create MEMORY, MyISAM and InnoDB sample tables
  2. View table type with SHOW CREATE TABLE
  3. Insert rows into MEMORY table, then restart MySQL serve to check the data
  4. Test transaction controls (COMMIT, ROLLBACK) with MyISAM and InnoDB tables

- Process and Thread
  1. Study **ps** command options how to view MySQL process and threads.
  2. Connect three **mysql** clients to monitor the process and threads

- Server Memory
  1. View **mysqld** variables with SHOW VARIABLES to identify MySQL server memory allocation (MySQL buffer and cache memories size).
  2. View Query Cache Size (query_cache_size) and change it as 64KB (65536) with SET command.

2. Backup and Restore

- Back up and restore database using mysqldump
  1. Use mysqldump to back up a database into a text file
  2. Use mysql to restore the database from the back up file

- Back up and restore table using SELECT … INTO OUTFILE / LOAD DATA INFILE
  1. Use SELECT … INTO OUTFILE to back up a table into a text file
  2. Use LOAD DATA INFILE to restore the table from the back up file

3. User Management

- Create a user “scott” with password “tiger” in localhost

- Grant all object with SELECT, INSERT, UPDATE privileges to scott@localhost

- Execute SELECT, INSERT, UPDATE to confirm privileges are given to scott@localhost

- Execute DEELTE to confirm privileges are NOT given to scott@localhost

- Revoke all privileges from scott@localhost

4. Configuration File

- Copy sample from /usr/local/mysql/support-files/ my-small.cnf file to /etc/my.cnf.

- Change following configurations in my.cnf and restart MySQL server to view the changes reflected with SHOW VARIABLES command.

Table 36 - my.cng configuration setting

Parameter	Default	Change to	Description
query_cache_size	0	256KB	Query Cache Size
max_connections	100	3	Maximum client connections

- Create a table with huge volume of data to query same SQL several times to find out how much cache are effective.

- Try to connect with four mysql from terminal to find out the connection of fourth mysql will be rejected from MySQL serve due to exceeding of the maximum number of connections.

5. Log Files

- Configure error log with my.cnf and view the log file.
  Insert an error purposely in my.cnf and view the error log after restarting MySQL server. MySQL server won't be able to start and the error should be captured in the error log file. Please make sure to restore original my.cnf (without the error) after this practice.

- Configure slow queries log with my.cnf and view the log file.
  (set **long_query_time** as 1 second and come up with a query takes more than 1 second to record the log)

  Hint: Cross joining tables such as t1, t2 without WEHERE condition may take a long time to process because it returns number of rows of t1 times t2.

- Configure general query log with my.cnf and view the log file.

- Configure binary log with my.cnf and view the log file with **mysqlbinlog** command. (Use -start-datetime, stop-datetime, start-position, --stop-position, --offset options to filter to view the binary log)

  Execute some DML such as INSERT, UPDATE and DELETE to view the binary log file.

Exercise 5 - phpMyAdmin (Web-based MySQL Admin Tool)

phpMyAdmin is one of the most popular Web-based MySQL administration tools especially in PHP developers community. The tool provides functionalities to execute any SQL statements such as DDL, DML and DCL and executes MySQL commands such a mysqldump. The tool allows MySQL administrators to access through the Internet (HTTP) to manage their MySQL servers.

The phpMyAdmin Project site:
http://www.phpmyadmin.net/home_page/index.php

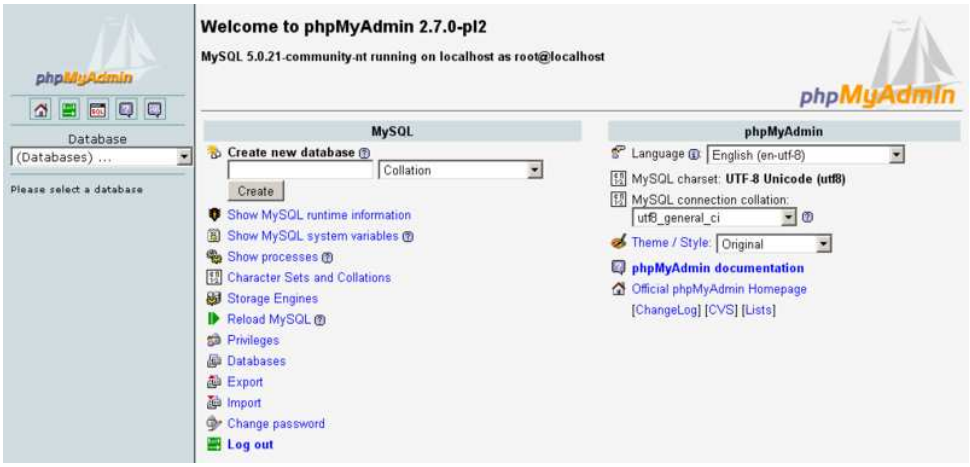


Figure 54 - phpMyAdmin Look & Feel

The objective of this exercise is to prepare PHP environment and to install phpMyAdmin, then go around a quick tour to find out features of the tool.

1. Preparation of Apache HTTP Server and PHP environment

- Apache Installation
  - Install Apache HTTP Server 2.2 from YaST

- PHP Installation
  - Install PHP 5 from YaST

- Apache2-php5 Installation
  - Apache2-mod_php5

- PHP Configuration (php.ini)

  Create a folder (/etc/php5/ or C:\PHP) and extract php-5.2.X-Win32.zip file.

  Note: This is for Windows environment; please follow your lecturer instruction for SUSE environment.

```
doc_root ="/srv/www/htdocs"
(doc_root ="C:/Program Files/Apache Software Foundation/Apache2.2/htdocs")

extention_dir=”/etc/php5/ext/”
(extention_dir=”C:/php/ext/”)

extention=php_mbstring.dll
extention=php_mysql.dll
extension=php_mcrypt.dll

Note: Remove “;” (comment) to activate setting
```

- Copy libmysql.dll and libmcrypt.dll files to C:\Windows\system32

  Copy C:\PHP\libmysql.dll and C:\PHP\libmcrypt.dll files to C:\Windows\system32

  Note: This is only for Windows environment

- Apache HTTP Server Configuration (Edit httpd.conf file)

  Update httpd.conf under /{Apache Home}/conf/ directory as follows:

  Note: This is for Windows environment; please follow your lecturer instruction for SUSE environment.

```
LoadModule php5_module C:/PHP/php5apache2_2.dll

AddType application/x-httpd-php .php
DirectoryIndex index.php index.html
PHPIniDir "/etc/php5"
(PHPIniDir "C:/php")
```

- Create test.php

  Create test.php as follows under the document root of the Apache HTTP Server

```
<?php phpinfo(); ?>
```

- Check PHP information from Web browser

System	Windows NT TSUTOMUONO-PC 6.0 build 6000
Build Date	May 31 2007 09:36:39
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:¥Windows
Loaded Configuration File	C:¥php¥php.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, data, http, ftp, compress.zlib
Registered Stream Socket Transports	tcp, udp
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, zlib.*

Figure 55 - phpinfo() at http://localhost/test.php

2. phpMyAdmin Installation

- Unzip phpMyAdmin-2.11.4-english.zip and rename the extracted folder as “phpMyAdmin”

- Move “phpMyAdmin” folder to /{Apache Home}/htdocs/
- Installation from Yast
  - ✧ Php_mystring
  - ✧ Php_mysql
  - ✧ Php_mcrypt

- Copy “config.sample.inc.php” to “config.inc.php” and update the file as follows:

```
$cfg['blowfish_secret'] = 'MySQL root password here';
/* YOU MUST FILL IN THIS FOR COOKIE AUTH! */

/* YOU MUST ADD FOLLOWING CONFIG! */
$cfg['Servers'][$i]['user'] = 'root';
$cfg[‘Servers’][$i][‘password’] = ‘MySQL root password here’;
```

- Access to http://localhost/phpMyAdmin/

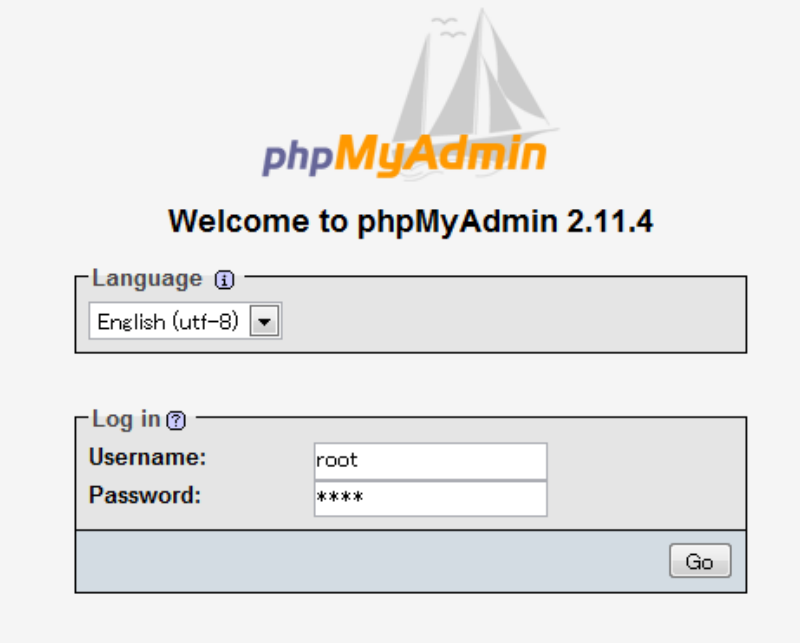


Figure 56 - Welcome to phpMyAdmin

Are you ready? Now let’s go for a quick tour of phpMyAdmin!

3. phpMyAdmin Quick Tour

- Login and Logout /Navigation Buttons

  Please read documentation and practice by yourself

Figure 57 - phpMyAdmin Navigation Buttons

- System Views

  Please check MySQL server information from phpMyAdmin Home page

Figure 58 - MySQL Server information

- Create database

  Please read documentation and practice by yourself

- Create table

  Please read documentation and practice by yourself

- Data manipulations (INSERT, SELECT, UPDATE and DELETE)

  Please read documentation and practice by yourself

- Export database

  Please read documentation and practice by yourself

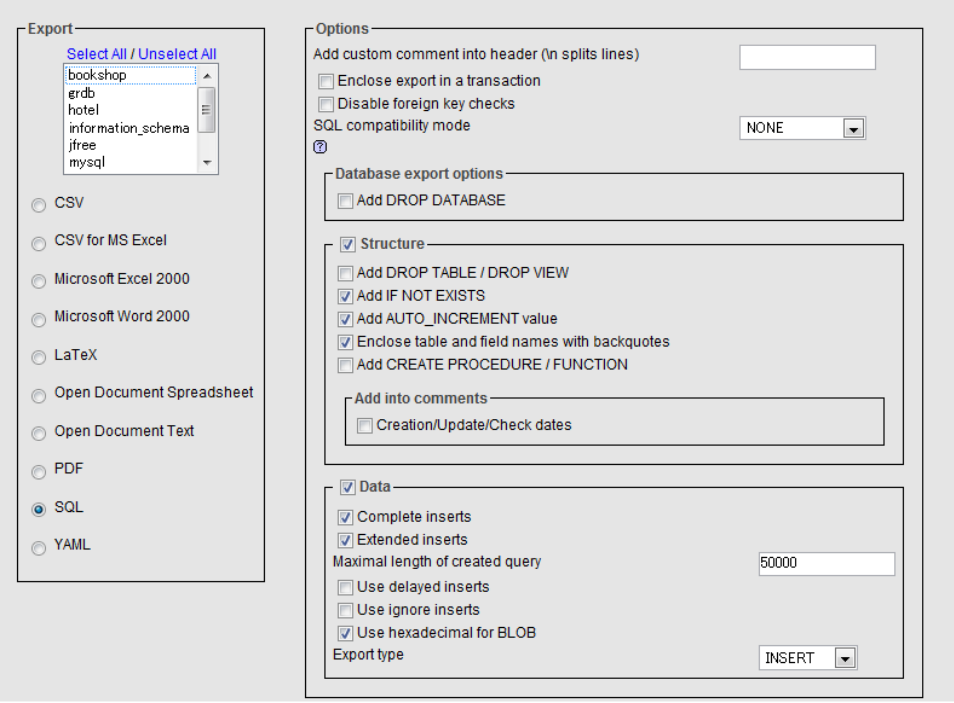


Figure 59 - Export database

- Import database

  Please read documentation and practice by yourself

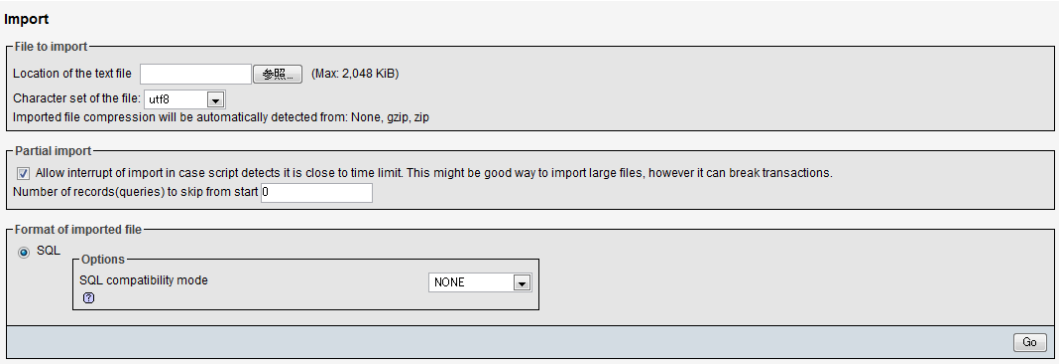


Figure 60 - Import Database

Tables and Figures

Figures

Tables

Indexes

Keywords