

DB 설계 및 디자인 – 2024 년도 2 학기

Group Project Guidelines

You are expected to form a group of 5 (or less) people to conduct a project during the semester. Each group is to identify a real world database development project. You are required to analyze, design, and implement a prototype database system using Oracle for your identified project.

You need to submit four project reports, which include three milestone reports - (a) Proposal, (b) Conceptual Model, and (c) Relational Model – and (d) Final Documentation (Final Project Report). At the end of the semester you will need to do the presentation/ demonstration of your project and database system. The submission deadline for each of the milestone/final reports is 11:59PM on each due date. Due dates will be announced through **LMS** or in class.

Project reports document the process and the results of your group project. Guidelines for each report/deliverable are as follows:

A. Project Proposal (프로젝트 제안서)

- (1) General description of the company where the group project is being conducted, which includes: industry, business goals of the company, products/services of the company, etc. (프로젝트를 행할 회사에 대한 설명: 산업, 비즈니스 목표, 회사의 제품/서비스 등)
- (2) Identify and report the purpose of the database system (and brief user requirements, if necessary) for your group project: why the organization develops a new database (and database system) (프로젝트 회사가 새로운 데이터베이스 (시스템)을 필요로 하는 이유. 어떤 목적인지; 필요할 경우, 설명하는 과정에서 간단한 user requirements 를 설명 – user requirements 를 너무 자세히는 필요없고 단지 왜 이 프로젝트가 그 회사에 필요하고 도움이 되는지를 설명하는 수준에서 도움이 될 만큼만 적을것.)

- (3) 11pts, Single space, no more than but close to 2 pages (폰트 사이즈: 11, Single Space, 되도록 2 페이지 전후로 정리)
- (4) Each team member provides an idea from work, study, and daily life. And pick your best topic idea. You can submit multiple ideas and tell me what your best idea is - please give a detailed description of each idea if it is the case. Briefly explain why you choose this idea and why the idea is reasonable (and feasible) for the project. Previous examples include: a company's database system, student club information systems, church member management system, the bookstore management system, election management system etc. (여러 개의 프로젝트 아이디어를 제출하여도 되나, 그 중 가장 베스트로 고려중인 것이 무엇인지 밝힐 것. 여러 개의 아이디어일 경우 각 토픽 모두 충분한 정보를 제공할 것. 제안서에 그 토픽이 왜 프로젝트를 위하여 합리적이고 가능한 토픽인지를 설명할 것.)

B. Conceptual Data Modeling

- (1) Describe all the details of user requirements (and business rules) as well as the questions to be asked to the database, and draw Entity-relationship diagram based on the requirements (and rules) described. (user requirements (사용자 요구사항) 과 business rules (그 기업 혹은 조직에서의 비즈니스 룰)들의 모든 세부사항들을 적고, 완성된 DB 에 어떠한 회사의 운영에 도움이 되는 질문들이 물어봐질 것인지를 적고, user requirements/business rules 에 기반하여 ER diagram 을 작성하시오.)
- So in this submission, you have to submit your complete (다시 말해, 이번 마일스톤 2 에서 제출해야 할 것은):
- (a) **User Requirements/Business Rules** (b) Important questions to be asked to DB and (c) **ER model**.
- (2) If your design should be different from user requirements, please explain why with reasonable reasons or assumptions. If it is not reasonable, you may lose points. (만약 ER 모델이 user requirement (혹은 business rule)과 다를 경우, 반드시 그 이유를 설명하시오. 그렇지 않을 경우에는 감점처리 됨)

- (3) Your final implementation of database in Oracle will need to maintain more than (at least) 5 tables (please check the requirements of implementation for details), so you need to have more than 5 entity types (최소 다섯 개 이상의 entity type) (note: subtype entity types or weak entity types, entity types derived from multi-valued attributes are not going to be counted as a separate entity type) and each entity type should have a reasonable number of attributes (at least 5 or more meaningful attributes – 최소 다섯 개 이상의 의미 있는 attributes) – otherwise, having the entity type in the model wouldn't be justified and lose points. So think accordingly in this design stage – in other words, the project scope cannot be too small and needs to be large enough to meet the requirement. If the conceptual model doesn't meet the requirements, you would need to resubmit. If the model is too poor, you may lose points so please come up with a good model.
- (4) To receive a good score (not only for the conceptual modeling but also for your entire project), you better need to use all or most of the ER and EER constructs we've discussed in class. So please try to come up with good, detailed and large enough requirements/business rules as well as the ER model so you can show your understandings on the various ER/EER constructs as well as the database designs steps.

C. Logical Database Design

- (1) Convert Entity-relationship data model into relational data model – at least in the third normal form. (please explain why it is in the third normal form)
- (2) Please follow the conversion rules as well as the normalization rules we discuss in class (for your explanation).
- (3) If you have suggested to improve your ER model and business rules/user requirements, you need to submit the relational model based on the improved/updated ER model.
- (4) In the submitted document, please also provide (1) the user requirements/business rules and (2) the ER model together with (3) the relational model, so I can compare your relational model with your ER model (and the descriptions) you submitted. If you have made any changes to your previous conceptual model or user

requirements/business rules, please submit the improved models/description (and briefly specify the change(s) and explain why).

D. Implementation in Oracle DBMS

- (1) Implement all relations specified in the relational data model (logical schema) developed in the logical database design phase, constructing tables in Oracle. The number of tables in your database needs to be more than 5 (since you must have converted from more than 5 strong entity types in ER).
- (2) Populate all tables in the database with sample data. Each table should have at least 15 records (instances). If you don't use the real data, please populate the fake data for the testing purpose.
- (3) Implement relevant constraints (e.g., primary key, foreign key, not null, etc.)
- (4) Create at least 10 useful, frequently used SQL queries. You need to explain why each of the queries is useful for what purposes. Out of the (more than) 10 queries, **at least 5 (or more) queries should involve two or more than two tables** using join operation, sub queries, etc. Therefore, please try to carefully think of the queries by thinking of its goal first (not the easiness to design the query, which can be easily seen) to receive a good score.
- (5) This is a project where you need to show your good understandings of what we've discussed in our class. So, in your queries, you need to **use as many SQL components we discussed as possible, rather than just using same and/or simple SQL statement logic/design repeatedly**. For example, just using normal select statement using a single table for many queries would not receive a good score.
- (6) It is your responsibility to test your implementation before the submission. Please make sure the database built using your codes doesn't have any problem. If your DB is not running when I grade, you will lose significant points.

E. Final Documentation (Final Report) Submission

The final report should be well organized summarizing your entire project progresses and details. **You should submit only one file with all the information, well organized.** Please properly number each step so it can be a well-organized report. The

report should have at least the following items (your report doesn't have to follow the numbering of the below items).

- (1) Group members and their roles.
- (2) Describe the project organization and its purpose (or business scenario) with the database of your design/development (e.g., what is the goal, why the database is needed, how it helps, etc.).
- (3) User/Data requirements and business rules.
- (4) Conceptual design: An (Enhanced) Entity-Relationship Diagram.
- (5) Logical design: Relation schemas, at least in the third normal form. You must use the textual notation, not the graphical notation.
- (6) Physical Design/Data dictionary: Tables with the description of each table (or relation) and its attributes, as well as their constraints (if any). For each attribute of a relation, the data dictionary needs to include its name, short description, data type/length, and legal value range (if appropriate), etc. You can use the format of the table in Lab1 document (available on Blackboard). For example, if you have a student table, you can create a table such as,

Table Name: Student

Attribute	Description	Data Type	Constraint
Uid	Student Number	VARCHAR2(4)	NOT NULL
Name	Student Name	VARCHAR2(30)	NOT NULL
Stuatus	Status	VARCHAR2(30)	
Major	Major	VARCHAR2(30)	

The format of the table can be a little varied depending on the information you want to provide, but your tables should have at least the information in the table above.

- (7) (Optional) Any additional assumptions or explanations supplementing your design and/or the data requirements.
- (8) SQL Statements (DDL): all SQL statements to create tables/constraints and populate data
- (9) SQL Statements (DML): ten or more useful SQL queries that meet the goal of your project DB. For each SQL statement, you need to explain what it does answer and why the SQL statement is important and useful. As stated already in Implementation section, at least five queries should involve two or more tables

using join or subquery (please refer to the implementation section again for the detail).

- (10) System manual: I will test your queries (or system) based on your manual so, please let me know if there is anything that I need to know to run your DB.

F. Final presentation and demonstration (If changed, re-announced later again)

- (1) Each group is given 12-15 min. to present its project to class and demonstrate its database implementation (such as Design, Tables created and their structure, execution of SQL queries, forms/reports, application system (optional), etc.).
- (2) In principle, all group members are required to attend and/or participate in the presentation/demonstration. Your participation would be also graded.
- (3) Group presentations start at the class time on the designated presentation day(s). The order of the presentations will be by the group number and the day will be announced before the presentation dates.

Grading Criteria:

Project proposal	10 Point
Conceptual Design	20 Points
Logical Design	20 Points
Prototype System and Documentation	30 Points
Presentation/Demonstration	20 Points

** Please let me know if you have any further questions on the project in class or by email (ijkim@mju.ac.kr). Thanks.