

👏👏👏👏 Congratulations. You did it !! This is an excellent submission  
👏👏👏👏

I am thrilled to inform you that you have completed all the requirements of the Designing an HR database project and earned a new milestone. 🏆🏆

Your work throughout the project has been exceptional and I was truly impressed to see the results of all your learning come together so clearly in this project. 🧐👏

The quality of your work is a testament to your dedication and commitment to mastering your skill in Database Architecture, Design and Modelling 🧐🧐🧐

Excellent work on addressing standout suggestions 🧐🧐

Keep up the great work, and I look forward to seeing what you accomplish next. 🔥

All the best 👍👍



✔ Translate a non-technical proposal into a technical proposal document.

Reviewer Note

Congratulations! You have done an excellent job understanding the business requirements and proposing a database design that meets those requirements. ✔

Changes since last review

You have correctly explained about **data ingestion method**, **scalability** and **flexibility** considerations 🌟 🌟

You have shown a great skill of understanding and listing out Business and Technical Requirements 🧑‍🎓 🧑‍🎓

Further Reading

- [Database Objects in DBMS](#)
- [Designing Highly Scalable Database Architectures](#)
- [Backup scheduling best practices to ensure availability](#)

- Complete "Data Architect Technical Requirement" section in step 1 in the starter template
- Provide **at least 2** justifications for creating a database
- Define data elements to be stored
- List database objects to be created (Students may wish to return to database objects section after completion of logical ERD)
- Define proposed data ingestion method
- Define who has data ownership
- Define user access recommendations
- List **at least 2 examples** of considerations taken to ensure data scalability and flexibility and provide an explanation
- Defined proposed storage method and provided an explanation
- Identify data retention requirements
- Propose a backup schedule and provide an explanation

✔ Gather business requirements for a new database request and create a non-technical proposal document.

Reviewer Note

Great job! I can see that you have completed the initial stage of identifying the business requirements for creating the database. ✔

Key Points

- When designing a database, it is important to consider both the business requirements and the data architecture to ensure that the database meets the needs of the organization. This involves identifying the data objects that need to be stored, how the data will be organized and indexed, and how the data will be accessed and maintained.
- To create an effective database design, it is essential to prepare a Business Requirement Document (BRD), which outlines the business requirements and identifies the data objects that need to be stored. The BRD should also include a Data Architecture Document (DAD), which details the technical aspects of the database, such as the schema, data types, and relationships between tables.
- Overall, by considering both the business requirements and the data architecture, you can create an effective database design that meets the needs of the organization and provides a solid foundation for data storage and management.

Further Reading

- [Why Business and Functional Requirements are Vital for a Project's Success](#)
- [Business Requirements Analysis](#)

- Complete "Data Architect Business Requirement" section in step 1 in the starter template
- Identify the business purpose for creating the database
- Outline data to be stored
- List estimate size of the database and growth rate
- Identify who will own/manage data
- Identify who will be able to access the data
- Identify sensitive/restricted data
- Outline data retention and backup requirements

## Relational Database Design

### ✔ Develop a logical ERD using Lucidchart.

#### Reviewer Note

Congratulations! I can see that you have correctly created the Logical ERD. I'm impressed with your work! 🍌 🍌

#### What are some common challenges when implementing a Logical ERD?

Implementing a Logical ERD can be a challenging task, and there are several common challenges that you may encounter. Here are a few examples:

1. Transforming the ERD into a physical database: One of the most significant challenges is transforming the Logical ERD into a physical database. This involves defining data types, indexes, and constraints, which can be a complex process.
2. Ensuring data consistency: Data consistency is critical to maintaining the integrity of the database. Ensuring that data is consistent across all tables and fields can be challenging, especially when dealing with large datasets.
3. Dealing with data redundancy: Data redundancy occurs when the same data is stored in multiple tables. This can lead to inconsistencies and errors in the database. Avoiding data redundancy can be challenging, especially when dealing with complex data models.
4. Managing database performance: As the database grows, performance can become an issue. Ensuring that the database is optimized for performance can be a challenging task, especially when dealing with large datasets.
5. Data security and access control: Managing data security and access control is critical to protecting the privacy and integrity of the data. Ensuring that only authorized users have access to the data can be a challenging task, especially when dealing with multiple users and roles.

By being aware of these common challenges, you can better prepare for them and develop strategies to mitigate their impact on the database implementation process.

- Complete the "ERD logical" section in step 2 in the starter template
- Normalize the data to the 3NF
- Create an entity for each table
- List Attributes
- Add relationship lines connecting entities
- Follow the visual requirements listed in the instructions
- Use Lucidchart's built-in template for DBMS ED Diagram UML
- Entity and attribute names can still be plain English

### ✔ Develop a physical ERD using Lucidchart

#### Reviewer Note

Congratulations! I can see that you have correctly created the Physical ERD. You have done an excellent job! 🍌 🍌

#### Common mistakes to avoid when creating Physical ERDs

Creating Physical ERDs can be a challenging task, and there are several common mistakes that you should avoid. Here are some examples:

1. Using inappropriate data types: One of the most common mistakes is using inappropriate data types for attributes. Using the wrong data type can lead to data integrity issues and performance problems.
2. Not optimizing for performance: Failing to optimize the database for performance can lead to slow queries and poor database performance. It is essential to consider the database structure, indexing, and query optimization when creating the Physical ERD.
3. Not enforcing data integrity: Failing to enforce data integrity can lead to data inconsistencies and inaccuracies. It is important to use constraints, such as unique, not null, and foreign key constraints, to ensure data integrity.
4. Ignoring normalization principles: Failing to follow normalization principles can lead to data redundancies, inconsistencies, and anomalies. It is essential to ensure that all entities obey the third normal form (3NF) to maintain data integrity and consistency.
5. Not considering scalability: Failing to consider scalability can lead to performance issues as the database grows. It is important to design the database with scalability in mind and consider the potential growth of the database.
6. Failing to document the ERD: Failing to document the Physical ERD can lead to confusion and misunderstandings. It is important to document the database design, including the schema, data types, and constraints, to ensure that everyone involved in the project understands the database structure.

By being aware of these common mistakes, you can better prepare for them and develop strategies to avoid them. This will help you create an accurate and effective Physical ERD that meets your business requirements.

- Complete the "ERD physical" section in step 2 in the starter template
- Tables and attributes should be given database friendly names now (think underscore or camel case)
- Attribute data types need to be defined
- Primary keys should be bold
- Relationship lines need to line up with PK / FK pairings
- Cardinality is required on this ERD
- Follow the visual requirements listed in the instructions

#### Reviewer Note

### You have correctly created the **Conceptual ERD** 🍌 🍌

ERDs are an essential tool for database designers as they help to visualize the relationships between entities and ensure that the database is well-structured and easy to use.

#### How can I ensure that I have identified all necessary entities?

Identifying all necessary entities is an essential step in designing an ERD. Here are a few tips to help you ensure that you have identified all the necessary entities:

1. **Understand the business requirements:** Understanding the business requirements is crucial to identifying all relevant entities. Make sure you have a clear understanding of the business processes, goals, and objectives before starting the ERD design.
2. **Conduct interviews and gather input:** Conduct interviews with stakeholders, subject matter experts, and end-users to gather input on the entities that should be included in the ERD. This can help you to identify any entities that may have been overlooked.
3. **Review existing documentation:** Review any existing documentation, such as business process diagrams, data flow diagrams, or database schemas, to identify entities that may have already been identified.
4. **Break down complex entities:** Sometimes, entities can be too complex, making it difficult to identify all the necessary attributes. Breaking down complex entities into smaller, more manageable entities can help ensure that you have identified all necessary entities.
5. **Look for relationships:** Identify all relationships between entities and use them to help identify any additional entities that may be necessary.

By following these tips, you can ensure that you have identified all necessary entities, which is essential for creating an accurate and effective ERD.

#### Further Reading

I recommend checking out the resources provided below. They will provide you with more information about ERDs and help you improve your skills in this area -

- [Introduction to ERDs - LucidChart](#)
- [Conceptual Modelling using the Entity-Relationship Model](#)

- Complete the "ERD conceptual" section in step 2 in the starter template
- Create at least 3 objects and show their relationships through connection lines
- This should be a first step towards 3NF, so chose attributes that will likely become future tables
- Follow the visual requirements listed in the instructions
- Use Lucidchart's built-in template for DBMS ED Diagram UML
- No attributes should be named and Crows foot notation is not required
- Consider an entity for any secure / restricted data

## Create A Physical Database

- ✔ Populate the database and demonstrate a working database by completing CRUD commands. ^

### Reviewer Note

Great Job! 🌟🌟

I am glad to see that you have correctly provided all the **CRUD** statements ✔

- Complete the "CRUD" section in step 3 in the starter template
- Screen shots should be taken of all SQL commands showing **code and results**
- Following commands like update/delete/insert, run a `select *` on the table affected to show results

- ✔ Develop DDL code to create a database in a SQL environment. ^

### Reviewer Note

Amazing work ! 🍌🍌

I just wanted to congratulate you on successfully completing all the **Postgres DDL statements** to build tables. You did a fantastic job! 🌟

You have correctly specified Primary and Foreign keys, data types in the CREATE table statement. This is crucial for maintaining data integrity, and you nailed it! Keep up the good world  
✔

### Useful Resources

If you want to keep improving your skills in MySQL, check out these helpful resources:

- [MySQL DDL](#): A comprehensive guide to MySQL Data Definition Language (DDL) statements that enables users to create, modify, and delete database structures. This guide can help you to expand your understanding of DDL statements and hone your skills in writing efficient and effective SQL code.
  - [How To Write Better Queries](#): A tutorial that provides tips on how to write efficient and effective SQL queries. It covers various topics such as filtering, sorting, and grouping data. By following these tips, you can write better queries that are optimized for speed and accuracy, which is vital in data analysis.
- Complete the "DDL" section in step 3 in the starter template
  - Create scripts (`.sql` file) to build tables with attributes as defined in the physical ERD
  - Primary and foreign keys must be included in the code