Tech ABC Corp

IT Department
Best Practices Guide
Database Architecture

Security:

All database security at Tech ABC Corp is based at the user level. Each employee in the company has a domain authenticated username that they will use to access any database they have been authorized access to.

To restrict access to:

- *Database*: do not grant user access to the database.
- *Specific data in a database*: grant user access to all tables in the database, then revoke access to any tables holding restricted data.

Backups:

All database backup schedules should be set based on priority (Standard, Archived, Critical):

- *Standard*: Backup schedule is a full backup 1x per week.
- *Archive*: Backup schedule is a full backup 1x per month.
- *Critical*: Backup schedule is full backup 1x per week, incremental backup daily.

Storage:

All databases are given a standard partition of 1 GB by the server group. Ask users about the expected growth of data. Databases larger than or expected to exceed 10K rows of data in the next year should ask for a large partition space.

Databases are stored on spinning disk by default. In-memory storage is available, but only for data that requires higher level computations (advanced analytics, machine learning applications).

Data Ingestion:

Direct Feeds: If setting up a direct feed from another database, please ensure a functional username is created by IT security. This will ensure an expiring username does not cause a data flow error.

API: If working with API, please submit the API address and information to IT security for evaluation before proceeding.

ETL: ETL is the current best practice for working with flat files. If the flat file will be regularly updated, an automated ETL process can be set up.

UDACITY

PROJECT SPECIFICATION

Designing an HR database

Data Architecture Foundations

CRITERIA	MEETS SPECIFICATIONS
Gather business requirements for a new database request and create a non-technical proposal document.	• 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
Translate a non-technical proposal into a technical proposal document.	• 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

Relational Database Design

CRITERIA	MEETS SPECIFICATIONS
Develop a conceptual ERD using Lucidchart.	• 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
Develop a logical ERD using Lucidchart.	• 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	• 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

Create A Physical Database

CRITERIA	MEETS SPECIFICATIONS
Develop DDL code to create a database in a SQL environment.	• 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
Populate the database and demonstrate a working database by completing CRUD commands.	• 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

Step 1: Data Architecture Foundations

The Memo

You have received the below memo as an introduction to the HR department's database request.

> *Hi,*
>
> *Welcome to Tech ABC Corp. We are excited to have some new talent onboard. As you may already know, Tech ABC Corp has recently experienced a lot of growth. Our AI powered video game console WOPR has been hugely successful and as a result, our company has grown from 10 employees to 20 in only 6 months (and we are projecting a 20% growth a year for the next 5 years). We have also gro from our Dallas, Texas office, to 4 other locations nationwide: New York City, NY, San Francisco, CA, Minneapolis, MN, and Nashville, TN.*
>
> *While this growth is great, it is really starting to put a strain on our record keeping in HR. We current maintain all employee information on a shared spreadsheet. When HR consisted of only myself, managing everyone on an Excel spreadsheet was simple, but now that it is a shared document I am having serious reservations about data integrity and data security. If the wrong person got their har on the HR file, they would see the salaries of every employee in the company, all the way up to the president.*
>
> *After speaking with Jacob Lauber, the manager of IT, he suggested I put in a request to have my HR Excel file converted into a database. He suggested I reach out to you as I am told you have experien in designing and building databases. When you are building this, please keep in mind that I want ar employee with a domain login to be have read only access the database. I just don't want them hav access to salary information. That needs to be restricted to HR and management level employees or Management and HR employees should also be the only ones with write access. By our current estimates, 90% of users will be read only.*
>
> *I also want to make sure you know that am looking to turn my spreadsheet into a live database, one can input and edit information into. I am not really concerned with reporting capabilities at the moment. Since we are working with employee data we are required by federal regulations to maintc this data for at least 7 years; additionally, since this is considered business critical data, we need to make sure it gets backed up properly.*
>
> *As a final consideration. We would like to be able to connect with the payroll department's system ir the future. They maintain employee attendance and paid time off information. It would be nice if th two systems could interface in the future*
>
> *I am looking forward to working with you and seeing what kind of database you design for us.*
>
> *Thanks,*
> *Sarah Collins*
> *Head of HR*

Let's get started

As mentioned in the memo above, you are going to design and build an HR database for Tech AB Corp. However, before you can start building, you need to complete both a business and technica proposal form answering some important questions about the database you are going to build.

Refer to your project starter Step 1. There, you will find the questions you need to answer, as well additional documentation from Tech ABC Corp's IT department concerning best practices for dat design, data governance, and other concerns.

While creating documentation is never high up on most people's list of fun things to do, it is still v important, as oftentimes business user expectations may not line up with technical realities. Havi proposal document to fall back on can save you a lot of head aches.

Hints

Make sure to keep the Business Requirement answers in plain English. Try to limit technical jargo this can often be off-putting to business partners. Remember, as a data architect, you will often h

Step 2: Relational Database Design

This step is where you will go through the process of designing a new database for Tech ABC Corp department. Using the *HR dataset* provided, along with the requirements gathered in step one, yo going to develop a relational database set to the 3NF.

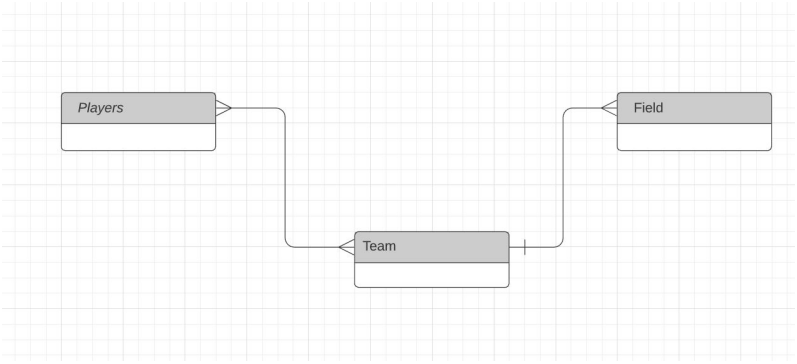
Using Lucidchart, you will create 3 entity relationship diagrams (ERDs) to show how you develope final design for your data.

You will submit a screen shot for each of the 3 ERDs you create. You will find detailed instructions developing each of the ERDs below and in the *starter template*.

Conceptual ERD

This is the most general level of data modeling. At the conceptual level, you should be thinking ab creating entities that represent business objects for the database. Think broadly here. Attributes column names) are not required at this point, but relationship lines are required (although Crow's notation is not needed at this level). Create at least three entities for this model - thinking about t 3NF will aid you in deciding the type of entities to create.

Use Lucidchart's built-in template for DBMS ED Diagram UML.

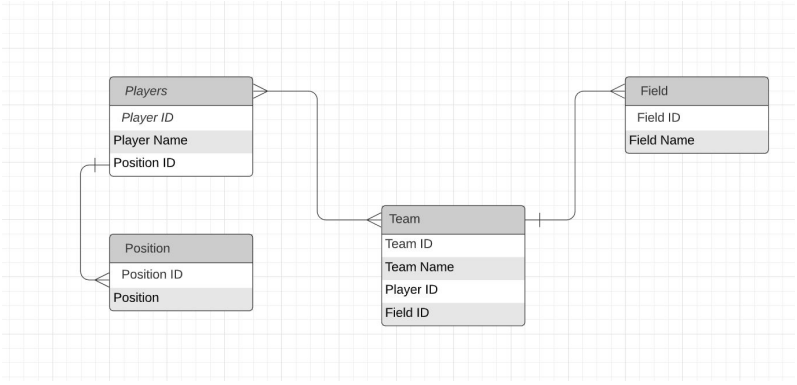


An example ERD after entity creation

Logical ERD

The logical model is the next level of refinement from the conceptual ERD. At this point, you shou have normalized the data to the 3NF. Attributes should also be listed now in the ERD. You can stil human-friendly entity and attribute names in the logical model, and while relationship lines are required, Crow's foot notation is still not needed at this point.

Use Lucidchart's built-in template for DBMS ED Diagram UML.



An example ERD with fields added

Physical ERD

The physical model is what will be built in the database. Each entity should represent a database complete with column names and data types. Primary keys and foreign keys should also be represented here. Primary keys should be in bold type with the (PK) designation following the fiel