

# Database Administration Project

## CNIT 48700, Fall 2024

### Goal:

The goal of this project is to create a database for a real-world setting and apply database administration principles to the created database.

This project is related to all the three learning outcomes of the course (understanding DBMS architecture, performing roles of a DBA including storage and user management, and applying methods of DBMS performance tuning).

### Tasks:

In this project, you will create a database and provide support for database administration.

You can create a university course enrollment system using the following files that have been provided (you will learn how to use these files for the project through each of the tasks that follow).

1. universityDB\_ER.pdf
2. universityDB\_create\_tables.sql
3. universityDB\_populate\_tables.sql
4. universityDB\_catalog.csv
5. universityDB\_object\_sizing.xls

You also have the option of creating your own database (using the provided files as an example) for another real-world problem of your interest (for example, for a DB developed as part of another course e.g., CNIT39200). However, the emphasis here is on providing database administration support and not just designing the database. Thus, while you may submit previous code to show how your system would function, you will be evaluated on the tasks outlined below:

### User Roles and Privileges

- Create users and user roles, and grant privileges as appropriate.
- Be creative: You have the freedom to add users, create roles and privileges befitting your database.

### Database Creation

If you choose to go ahead with the university course enrollment system,

- The ER diagram of the database has been provided in the file universityDB\_ER.pdf
- The SQL script to create the tables and indexes for the database has been provided in the file universityDB\_create\_tables.sql
- Using the SQL script as a starting point, create tables and *indexes* with appropriate tablespace and storage allocations.
  - Specify locations of the physical and logical storage structures
  - Compute the storage needed for the tablespace (the file universityDB\_object\_sizing.xls provides a template—you are free to change the suggested values).

## Data Insertion using SQL\*Loader

If you choose to go ahead with the university course enrollment system,

- Populate tables in the database using SQL\*Loader and the scripts provided in the file `universityDB_populate_tables.sql`.
- Test the database to make sure it is usable.
- Update dates to reflect 2023 and 2024 instead of 2007 and 2008, respectively.
- Create the catalog table, using SQL\*Loader and the provided catalog file (`universityDB_catalog.csv`).

## Backup and Recovery

- Simulate a scenario where your database needs to be recovered.
- Develop appropriate backup and recovery procedures utilizing tools of your choice. You may provide either online or offline backup mechanism.

## Benchmarking

- Create a representative set of queries.
- Generate query plans for each of the queries.
- Benchmark the performance of these queries by capturing server data and tools such AWR snapshot.

## Phase 1: Estimate Storage Requirements

Estimate the amount of disk space that will be required for a proposed *production* database, based on the following assumptions.

### Production Database Tables:

Low volatility:	<b>CAMPUS</b> (25 rows), <b>DEPARTMENT</b> (40), <b>COURSE</b> (2,000), <b>TERM</b> (20)
Medium volatility:	<b>COURSE_OFFERING</b> (10,000), <b>COURSE_SECTION</b> (25,000), <b>INSTRUCTOR</b> (750)
High volatility:	<b>STUDENT</b> (40,000), <b>TERM_ENROLLMENT</b> (250,000), <b>COURSE_ENROLLMENT</b> (1,000,000), <b>SECTION_ENROLLMENT</b> (2,500,000)

### Prototype Database Assumptions:

Tablespaces:	Separate tablespaces for tables and indexes based on volatility
Indexes:	100% of corresponding table size
Rollback:	One may choose to use Automatic Storage Management or manually manage the rollback segments. If one manually manages the rollback segments, use the following: Low volatility =N/A, medium=10% of data size, high=20% of data size
Fixed Header:	100 bytes/block
Variable Header:	2*23 bytes/block
<b>PCTFREE:</b>	Low volatility =5, medium=10, high=20
Block size:	Individual determines.

## Deliverables:

- **Phase 1 (Due October 10, in-class presentation)** (10 points)
  - Estimate storage requirements for a production Course Enrollment database using the Excel Template provided.
- **Phase 2 (Due November 5 and November 7, in-class presentation)** (30 points)
  - Create a prototype of the production database that can be used for application development.
  - Create tables and indexes with appropriate tablespace, PCTFREE and storage allocations.
  - Create development users and roles, then grant privileges as appropriate.
    - Demonstrate scenarios where different user roles attempt to access the database in unauthorized ways.
  - Populate the databases using the script files provided and SQL\*Loader or other utility/technique of your choice, then test the databases to make sure it is usable.
  - Document our databases using standard data dictionary views.
- **Phase 3 (Due December 3 and December 5, in-class presentation)** (30 points)
  - Develop database backup and recovery procedures.
    - Document your experience with RMAN
  - Document our databases using standard data dictionary views.
  - Enable auditing and check that the audits work as expected.
    - Benchmark query performance by capturing server data, tools such as AWR snapshot.
- **Final submission (Due 5:00 PM, December 7)** (30 points) Document the database with the enlisted tasks and submit a typed pdf document to the submission link on BrightSpace.