CS348 Project Writeup William Lin

* **Users**: Stores user information (user\_id, name, email, phone\_no).
* **Teams**: Information on teams within the organization (team\_id, name, description).
* **Tasks**: Tracks tasks (task\_id, title, description, status, due\_date, assigned\_to, oncall\_team, completion\_time, creation\_time).

Indices: Date, Status on Tasks

Supported Queries:

Date: Filter for overdue tasks ( date < current, status != completed), filter for incomplete tasks before certain date

Status: filter for incomplete, pending, in progress, overdue tasks

For a task management web app, Read Committed isolation is likely the best choice because:

Consistency: It avoids dirty reads, ensuring that users always see committed, consistent data.

Performance: It provides a good balance between consistency and performance.

Use Cases: Your app may allow users to edit, assign, or update tasks, so you don’t want them to read data that is in the process of being modified (which could happen with Read Uncommitted).

Lessons:

Create a good database schema at the beginning with thorough thought. It is hard to modify a database after production. Database migrations is very difficult.

Have flexible apis with multiple endpoints for different queries

Stored procedures are used to filter for tasks before certain dates, and filter for overdue tasks using raw sql. We also used the Django API that provides ORM access to database for easy data queries and CRUD actions. Django Api uses ORM to sanitize user input and make sure that no invalid data is written to the database.