Question 1:

- A **database system** is the combination of a database management system (DBMS) which creates and manages databases, the stored data itself, and the computer applications that use the database.
- A **canned transaction** is a standard pre-defined transaction (such as a query or update) that is frequently used by naive users.
- A **deductive database system** is a type of database system that uses deductive reasoning to create new information out of existing data, using rules and facts stored in the database.
- A **transaction-processing application** is a software system that handles multiple transactions concurrently and maintains consistency within the database, making sure that data can be processed in real-time without issues.
- **Three-tier architecture** describes how many modern database applications are separated into three tiers, with the application client as the frontend, connecting to the application server with the business logic, connecting to the database system to access the data.

Question 2:

The database approach describes a central database that can be accessed and modified by multiple applications and end users, and it is managed by a database management system (DBMS) that serves as an intermediary between users and the database, allowing users to create, read, update, and delete data in the database. Some important characteristics of the database approach is its self-describing nature (DBMS catalogs describe the constraints and structure of the database), program-data independence (programs are unaffected by changes in the database's structure), data abstraction (hiding data storage details), support of multiple views, and sharing data between users concurrently. In contrast, traditional file systems involve data stored in independent files without regard to whatever program needs the data. The file system runs into problems with data redundancy, as the same files are required by multiple programs, which leads to inconsistencies in the data due to numerous changes from various program sources. In this system, the program and the data are dependent on each other, as the program is written based on the shape of data. Additionally, without centralized control like the database system, data sharing between application programs is difficult and limited.

Question 3:

Controlled redundancy is when duplicate data is intentionally created and managed to improve performance and simplify retrieval. In this case, the DBMS ensures that the duplicate data is consistent with each other no matter its location. An example is a system backup - where the entire database is duplicated to ensure that the data is preserved in the case of any accident.

On the other hand, uncontrolled redundancy occurs when a database has duplicate data that is created on accident and/or poorly managed. This duplicate data often leads to inconsistencies, where the same information is stored in multiple places, but are not coordinated to update simultaneously. For instance, there could be two separate tables titled COURSE_STUDENTS and SECTION_STUDENTS, both containing information of each student's STUDENT_ID. If a student decides to drop a class, and the student is removed from COURSE_STUDENTS, but SECTION_STUDENTS is not updated accordingly, then the redundancy is not controlled and can cause issues.

Question 4:
The relational data model organizes data into tables (also called relations), and the relationship between the tables is established using keys. That makes this model useful for general purpose databases, as it allows for easy access in grabbing related data. The object data model represents data as objects, where it represents an instance of a class with properties and methods. This makes this model useful for object-oriented applications with complex data processing, though a pure object-oriented approach is rarely used for database systems (they are usually combined with relational DBMSs). The XML data model combines the description of the data with its data values, represented as elements in a hierarchical tree with XML tags. Each element inside of a tag can have its own properties and elements nested inside of it. This makes this model useful for storing information and exchanging data between systems, which is why it is commonly used in web applications (such as e-commerce). In summary, the key difference between the three models is how it represents data: either through tables (relational data model), objects (object data model), or elements (XML data model).

Question 5:
A procedural DML requires a user to specify what kind of data they need, and how they get the data. On the other hand, a non-procedural DML, also called a declarative DML, only requires a user to specify what kind of data they need, but they do not need to specify how to get the data. This makes nonprocedural DMLs easier to learn and use, since the database system, not the user, is responsible for figuring out how to access the data.

Question 6:
Different users that would use the database shown are students, faculty, registrars, and IT support.

- Students are naive/parametric users. Applications they would need include: A course registration system to select and enroll in new courses, a gradebook system to view their grades for each course, and a course planning system to ensure that they will graduate on time by taking all of their required classes in an order that satisfies prerequisites. Students would need a graphic user interface that allows them to view their course schedule, gradebook, and transcript. They also need a GUI interface for course registration that allows them to input and select courses to take for any given semester.

- Faculty are also naive/parametric users. Applications they would need include: A gradebook system to enter students' grades, and a course section system to view a roster for each section they teach. Faculty would need a forms-based interface to enter student grades and a graphic user interface to view each course section.
- Registrars are casual users. Applications they would need include: A course registration system to confirm student registrations and modify when necessary, a course catalog system to confirm the list of courses available, and a course section system to assign professors, times, and locations for each section. Registrars would need a forms-based interface to enter course and section information, as well as a graphical user interface to manage student enrollment and courses.
- IT support are operations and maintenance personnel. Applications they would need include: A security management system, and a report system to address any technical issues that users report. IT support would need a graphic user interface to address these technical issues and interact with the DBMS, as well as a command line interface for tasks that cannot be addressed in GUIs.