ADVANCED DATABASE SYSTEMS Coursework-Parts A & B

PRACTICAL GROUP ASSIGNMENT: DATA DEVELOPMENT

This is the **final** coursework. The whole coursework (parts A and B) counts towards 30% of the course, i.e. **3 marks in total**. Please note that you need to pass this.
Submission **deadline** 27/1/2023 at 5 p.m.

Part A

Deliverables:

1. Database Design (Table Specifications)
2. Evaluation (Test Plan)
3. Database Development (Script Files)

Database Development:

Nominate by email (subject: "ADS group nomination") to p.koukaras@ihu.edu.gr (cc c.tjortjis@ihu.edu.gr) groups of 2, by 6 p.m. 05/12/2022. If you cannot nominate a group, you will be assigned to one.

Database Description:

Below are the business rules for **an apartment rental database**. Read them through and complete the requirements topic below.

1. A company wants to commission a database for recording the owners, the apartments, the tenants the rentals and the reviews of each apartment. All apartments are based in Greece.
2. The owner can have many apartments. Details like name, surname, email, city, date of birth etc. are recorded. Each owner has a unique Tax Identification Number (TIN).
3. Each apartment has an owner and one owner only and unique ID. Other details that are needed are the street, street number, postal code, city, floor, price, etc.
4. Tenants have a unique ID and details such as the name, surname, gender, email, date of birth, country, number of tenants, etc. can be recorded. In case of a group of tenants, only one of them is needed to be recorded by the system with his/her details.
5. The rental must include details like the apartment ID, tenant ID, start date and end date for the rental. A rental ID is also required as the same apartment can be rented more than once by the same tenant. Alternatively, a composite key can be used (apartment_id, tenant_id, start_date).
6. The tenant can leave a review for the apartment. A review ID is needed as an apartment can receive many reviews, a tenant can leave multiple reviews, even for the same apartment (in case of more than one rental). The date of the review must also be recorded, along with a star review point and a review text.

Complete the following tasks:

Task 1: Table Specifications

Create a table (there is an example in Appendix A) with the database specifications based on the given Entity Relationship Diagram (ERD). You should consider the following:

- appropriate table name, datatypes, constraints and defaults
- integrity
- case

Test Plan (see end of file)

Document and evaluate all parts of the database, following the table shown at the end of this file. Show all your tables, your primary keys, the sequences, all your queries by recording the actual query, which element was tested (i.e. the tables of the database, query #12, query showing a projection of 3 columns, dropping all the tables of the database etc.), the expected result, whether the result was successful and a screenshot of the result.

Task 2: Script Files

Use the ERD and your table specifications to create the following script files (all files should have an appropriate extension with proper details, run commands and comments).

1. **create_*yourTwoSurnames*.txt (e.g. create_Black_White.txt)**
   - Create table commands for all tables
   - Tables should be created in the correct order to maintain integrity
   - Include:
     - NULL constraints and defaults
     - Sequences
2. **constraint_ *yourTwoSurnames*.txt**
   - Alter table… add constraint commands for all tables (primary keys, foreign keys, unique, checks)
   - Constraints should be created in the correct order to maintain integrity and named after the instructions on your lad notes
   - Any changes to the attributes you consider important (add new attributes, defaults, upper, change attributes, etc.)
3. **insert_ *yourTwoSurnames*.txt**
   - Insert commands for all tables
   - Regarding the number of entries per table, do not overload the database
   - Inserts should be in the correct order to maintain integrity
   - Data should be appropriate to support queries in task 2.5
   - Use two different methods for inserts (with or without the column list, be careful when you have used DEFAULT)
4. **drop_ *yourTwoSurnames*.txt**
   - Drop table, constraints and sequences
   - Drops should be in the correct order to maintain integrity (do not just drop the table)
   - Any additional commands you consider important
5. **query_ *yourTwoSurnames*.txt**
   - Queries to achieve the following data:
     a) All the records from one table
     b) A projection with 3 columns
        - data returned sorted in reverse alphabetical order
     c) Restriction queries with *multiple* clauses demonstrating the following
        - E.g. the tenants from a specific country
        - a condition matching a pattern e.g. begins with J, ends with SON.
        - a negative condition predicate e.g. are *not* from Greece
        - a date range condition e.g. last year April - September
     d) Join queries with data from
        - 2 tables
        - 3 tables
        - More tables
     e) Aggregate functions
        1. Show the tenants whose surname starts with A and ends with S.

2. Show the total number of tenants per country.
3. Which apartment was the most popular (was rented the most times)?
4. Show the owners (surname, name and TIN) in an ascending order and the apartments they own.
5. Which owner (surname, name and TIN) has the most apartments.
6. In which city you can find the most expensive apartment?
7. How many reviews were written in 2020?
8. Which tenant (surname and name) has left the most reviews?
9. Show the apartments that were rented in August (regardless of the year).
10. How many apartments that haven't been rented exist?
11. Show the oldest female tenant
12. Show the owners that none of their apartments were rented.
13. Which tenants rented an apartment that its street address has the word "SAINT" in it?
14. Show the apartments in descending order, based on the average of their review scores.
15. Which apartment in "THESSALONIKI" has the highest review average?
16. Which tenant has rented the most apartments?
17. Show the average score of the apartment reviews made by female and male tenants.
18. Show the most popular city (the city that its apartments were rented the most times) per gender
19. Assume a username for the tenants is made up of the first 4 letters of the first name added to the first 4 letters of the surname. Write the query to create it and show the username for each user. Do not attempt to input it in the tenants table.
20. Show the total revenue from ALL the apartments (total of the price multiplied per rental days) per year.

- Other complex queries you can think of.

Extra information and tasks:

- You should use **SQL and Oracle** to produce your database
- The database design is minimal (there are indeed improvements that can be made). The goal of the assignment is to test the ability to create a physical database based on a specific database design.
- Comment your scripts appropriately
- Be careful with the **date format (DD-MON-YYYY) ONLY**.
- All text inserts should be in **UPPER Case**
- Select - create carefully your primary keys.
- Do not insert many rows. Keep it, if possible, to less than 20 rows per table.
- Insert values that will return results (i.e., tenants from England, apartments containing the word "Saint" in their street name, etc.) and also repeated values in various columns (i.e. genre, surnames, country, etc)
- **At the end of each script show what you have done with the appropriate commands (show all tables, describe the tables, show all the sequences and constraints - all the pks, fks, checks, unique -, show all the data from the tables, show that the database is dropped).**

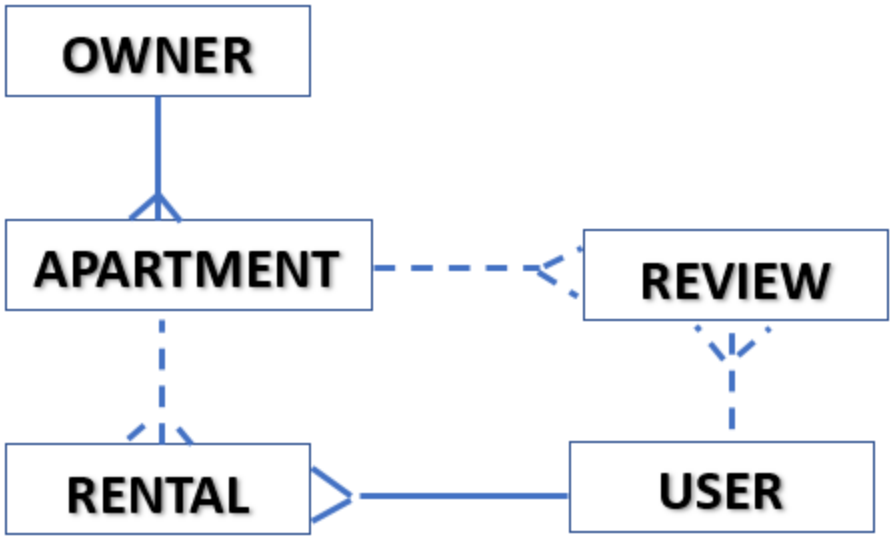
Appendix A: Sample Table Specification (example – incomplete)

ATTRIBUTE	DATATYPE	CONSTRAINT	DEFAULT	DESCRIPTION
owners				
TIN	NUMBER(9)	PK, pk_TIN		The Tax Insurance Number of the owner
o_lastname		UPPER, NOT NULL		
city	VARCHAR2(30)		"SPARTA"	
email		UNIQUE		
…				
tenants				
t_firstname	VARCHAR2(30)	UPPER		
contact_number				
gender	CHAR()	CHECK IN ("M", "F", "O")	"F"	
dob	DATE			Date of birth

ATTRIBUTE DESCRIPTION (NOT EVERY ATTRIBUTE)

Attribute	Description
Floor	0 for ground, 1 for 1st floor, and so on.
Price	Price per day in euros (with two decimal points)
Owner_Address	Can be written all together (i.e. 13 Aristotelous st.)
Tenant_Country	The country of the tenant's origin.
Number_of_tenants	How many people will be staying at the apartment
Stars	From 0 up to 5

INDICATIVE ERD



Test Plan

How to plan your testing and demonstrate your results

ID	Test	Element tested	Expected Result	Actual Result
1	SELECT * FROM TAB;	Database tables	7 tables returned	As expected
+SCREENSHOT				
2	SELECT DISTINCT CITY FROM OWNERS;	Query #33	ATHENS, SPARTA, THESSALONIKI, VERIA	As expected
+SCREENSHOT				
3	SELECE USERNAME FROM CUSTOMERS	Query #39	25 ROWS RETURNED	Not returned

```
SQL> SELECT username from customers;

USERNAME
-------------------------
CASM
MASM
PABR
PEHI
ALSM
JAFI
SAGR
JASM
LIWH
JAHA
JASM

USERNAME
-------------------------
ALHA
NIRI
CASM
MASM
PABR
PEHI
ALSM
JAFI
SAGR
JASM
LIWH

USERNAME
-------------------------
JAHA
JASM
ALHA

25 rows selected.
```

***TEST ID IS A DUMMY TEST AND SCREENSHOT (NOT FROM THE REQUIRED DATABASE)**

Part B

1. Write a report of up to 500 words, briefly discussing prominent indexing techniques. Propose suitable indexes for the database you created in Part A.

0.4 marks

2. Could a star or snowflake schema be used for the database you created in Part A? What changes should you introduce? Provide an implementation.

0.6 marks