**Final Exam**

Classroom: Online

Assignment Points: 25 points

**Name: John Wilson**

**Student ID: 107626362**

**Exam rules:**

* You MUST submit this final exam by **7/21/2022, 11:59 pm.** There will not be any extension or late submission.
* Submission: submit in **Canvas in pdf or word doc**.
* This is open book exam and any kind of resource materials are allowed.
* Collaborations and consultations are NOT allowed. Do your own work.

**Section 1: 5 points**

1. During normalization process, we tend end up with more tables than we started?

A. True

B. False

2. You should remove partial dependency of non-Key attributes on the entire key in which normalization form?

A. First Normal Form

B. Second Normal

C. Third Normal Form

D. Fourth Normal Form

3. Which one of the following is used to create/define table and alter the of the structure in relational database system?

A. DML (Data Manipulation Language)

B. DDL (Data Definition Language)

C. Query

D. Relational Schema

4. A composite primary key may have a null value so long as other candidate keys remain not null.

A. True

B. False

5. Which of the following clause is used to filter the conditions in the SQL statement?

A. THEN

B. WHILE

C. WHERE

D. IF

6. Which SQL keyword is used to sort the result-set?

A. ORDER BY

B. ORDER

C. SORT BY

D. SORT

7. The BETWEEN operator is inclusive: begin and end values are included.

A. True

B. False

8. \_\_\_\_\_\_\_\_\_LIKE\_\_\_\_\_\_\_\_\_\_\_ operator is used to match a specified pattern in a SQL statement.

9. \_\_\_\_\_\_\_\_Normalization\_\_\_\_\_\_\_\_\_\_\_\_\_ is the process to eliminate data redundancy and to make functional dependency of attributes.

10. The complete SQL statement may be complicated but it must include at least these two keywords: \_\_\_\_\_\_SELECT\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_FROM\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

11. Write a SQL statement to select the unique "LastName" column from the "Students" table?

SELECT UNIQUE LastName FROM Students;

12. Write a SQL statement to select all the records from a table named "Characters" where the ‘FirstName’ starts from ‘A’ or ‘B’.

SELECT \* FROM Characters WHERE FirstName LIKE ‘A%’ OR FirstName LIKE ‘B%’;

13. Write a SQL statement to select all the records from a table “Customers” where the “LastName” ends with “c”.

SELECT \* FROM Customers WHERE LastName LIKE ‘%c’;

14. DELETE FROM supplier statement does the following:

A. Delete all rows in the supplier table, including the table structure

B. Delete all rows in the supplier table

C. No data is deleted without WHERE clause

D. Delete the first row in the supplier table

15. In GROUP BY SQL statements, you remove groups that do not meet the conditions with the use of:

A. WHERE clause

B. HAVING clause after GROUP BY

C. WHERE clause first, and then use the HAVING clause

D. HAVING clause first, and then use the WHERE clause

**Section 2: 3 points**

Normalize the following form into **3NF.** Only your 3rd NF will be graded.

|  |  |  |
| --- | --- | --- |
| **University Departments Sample Form** | | |
|  |  |  |
| Dept Name | ……………………….. |  |
| Building Num | ……………………….. |  |
| Phone 1 | xxx-xxx-xxxx |  |
| Phone 2 | xxx-xxx-xxxx |  |
| Phone 3 | xxx-xxx-xxxx |  |
|  |  |  |
| Instructor Name | Subject | Gender |
| ……….. ……………. | …………….. | X |
| ……….. ……………. | …………….. | X |
| ……….. ……………. | …………….. | X |
| ……….. ……………. | …………….. | X |

This is the University departments sample form used by many departments.

Each department belongs on only one building and department may have multiple phone lines. Each instructor teaches only one subject and belongs on only one department.

If there is no many to many relation (no concatenated key), many times you can put directly into 3rd NF, i.e., do not carried away with unnecessary normalization. There is not always 1st, 2nd and 3rd NF needed. **Normalize**, as you did in HW-1 Normalization.

Hints: List all attributes.

Identify the repeating group of attributes.

Create entities and keys e.g. PK/FK.



**Section 3: 3 points**

Create **ERD design** for following scenario:

Your data model design (ERD) should include relationships between tables with primary keys, foreign keys, optionality and cardinality relationships. Captions are NOT required.

**Scenario:** There are 3 tables with 2 columns in each table:

**Department (** Dept ID, Department Name )

**Employee (** Employee ID, Employee Name )

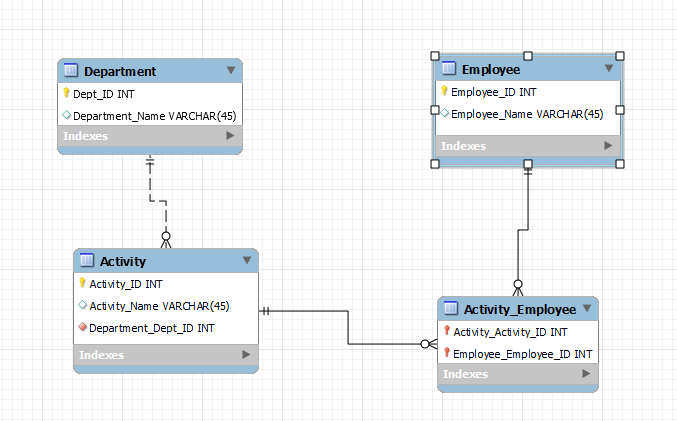
**Activity (** Activity ID, Activity Name )

Each Employee must belong to ONLY ONE Department.

Department may have ZERO, ONE OR MORE Employees, i.e. Department may exists without any employee.

Each Employee may participate in ZERO, ONE OR MORE Activities

Each Activity may be performed by ZERO, ONE OR MORE Employees.

****

**Section 4: 3 points**

1. Create table **T1** with following columns and constraints.

Note: DO NOT use alter table, list all constraints while creating table.

C1 INT (10) Primary key

C2 INT (10)

C3 INT (10)

C4 VARCHAR (40)

**Constraints:**

C3 NON-ZERO

C2 greater than C3

C4 default value of 'HR'

CREATE TABLE IF NOT EXISTS T1

(

C1 INT (10) NOT NULL PRIMARY KEY,

C2 INT (10),

C3 INT (10),

C4 VARCHAR (40) DEFAULT 'HR',

CONSTRAINT CHK\_4a CHECK (C3 <> 0 AND C2 > C3)

);

1. Create table **T2** with following columns and Foreign Key.

Note: DO NOT use alter table, create FK while creating table.

C5 INT (10) Primary key

C6 INT (10)

FK on C6 column referencing to C1 column in table T1 above.

CREATE TABLE IF NOT EXISTS T2

(

C5 INT (10) NOT NULL PRIMARY KEY,

C6 INT (10),

CONSTRAINT FK\_C1\_C6 FOREIGN KEY (C6)

REFERENCES T1(C1)

);

1. Explain, in short, the meaning and importance of Referential Integrity (RI).

Referential integrity is a database concept that guarantees that relationships between tables remain consistent therefore if a table, x, has a relationship with another table, y, and a row of x references a row of table y then that row of table y must exist. Referential integrity is importance to assure that data relationships are accurate and consistence across databases as well as across servers and networks.

**Section 5: 8 points**

All questions are based on below **Employee** table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EmpId** | **ManagerId** | **Name** | **Department** | **Salary** | **City** |
| 1 | 0 | Alex Smith | Admin | $90,000 | Boulder |
| 2 | 1 | Amy Mars | Admin | $50,000 | Longmont |
| 3 | 1 | Logan Mars | Admin | $70,000 | Longmont |
| 4 | 1 | James Mont | Marketing | $55,000 |  |
| 5 | 6 | John Smith | Marketing | $60,000 | Boulder |
| 6 | 1 | Lily Mars | Marketing | $95,000 |  |
| 7 | 6 | Ravi Grace | Database | $75,000 | Longmont |
| 8 | 6 | Tara Frank | Database | $80,000 | Longmont |
| 9 | 6 | Tom Ford | Database | $65,000 |  |
| 10 | 6 | William Cruze | Database | $85,000 | Longmont |

1. Write a SQL statement to find the Name and Salary who has **5th HIGHEST** Salary in the entire Employee table.

SELECT Name, Salary

FROM (SELECT Name, Salary FROM Employee ORDER BY Salary ASC LIMIT 5)

ORDER BY Salary DESC

LIMIT 1;

1. Write a SQL statement to find the Department and their count whose count is more than 3.

SELECT Department, COUNT(Department)

FROM Employee

GROUP BY Department

HAVING COUNT(Department) > 3;

1. Write a SQL statement to show Name, Department and City.

However, if City is NULL, then display 'Broomfield' otherwise display City itself.

SELECT Name, Department, City

FROM Employee

CASE WHEN City IS NULL THEN ‘Broomfield’

ELSE City

END;

1. Write a SQL statement to find distinct employee Name who is also a Manager

SELECT DISTINCT Name

FROM Employee

WHERE EmpId =

(SELECT DISTICT ManagerId FROM Employee WHERE ManagerId <> 0);

1. Write a SQL statement to find Maximum, Minimum and Average Salary from the entire Employee table.

SELECT MAX(Salary), MIN(Salary), AVG(Salary)

FROM Employee

;

1. Write a SQL statement to show Name, Department and Salary who earn MORE THAN the Average Salary in **THEIR department**. You must use sub-query.

SELECT Name, Department, Salary

FROM Employee e

WHERE Salary > (SELECT AVG(Salary) FROM Employee e2 WHERE e2.Department = e.Department);

1. Write a SQL statement to show Name, Department, Salary and their Rank **WITHIN Department** from highest to lowest salary.

i.e, Salary rank must reset and re-rank start from 1 for EACH Department.

SELECT Name, Department, Salary, RANK()

OVER (PARTITION BY Department ORDER BY Salary DESC)

AS Salary\_Rank

FROM Employee;

1. Write a SQL statement to find HIGHEST paying employee’s Name and Salary from the entire Employee table. You must use sub-query.

SELECT Name, Salary

FROM Employee

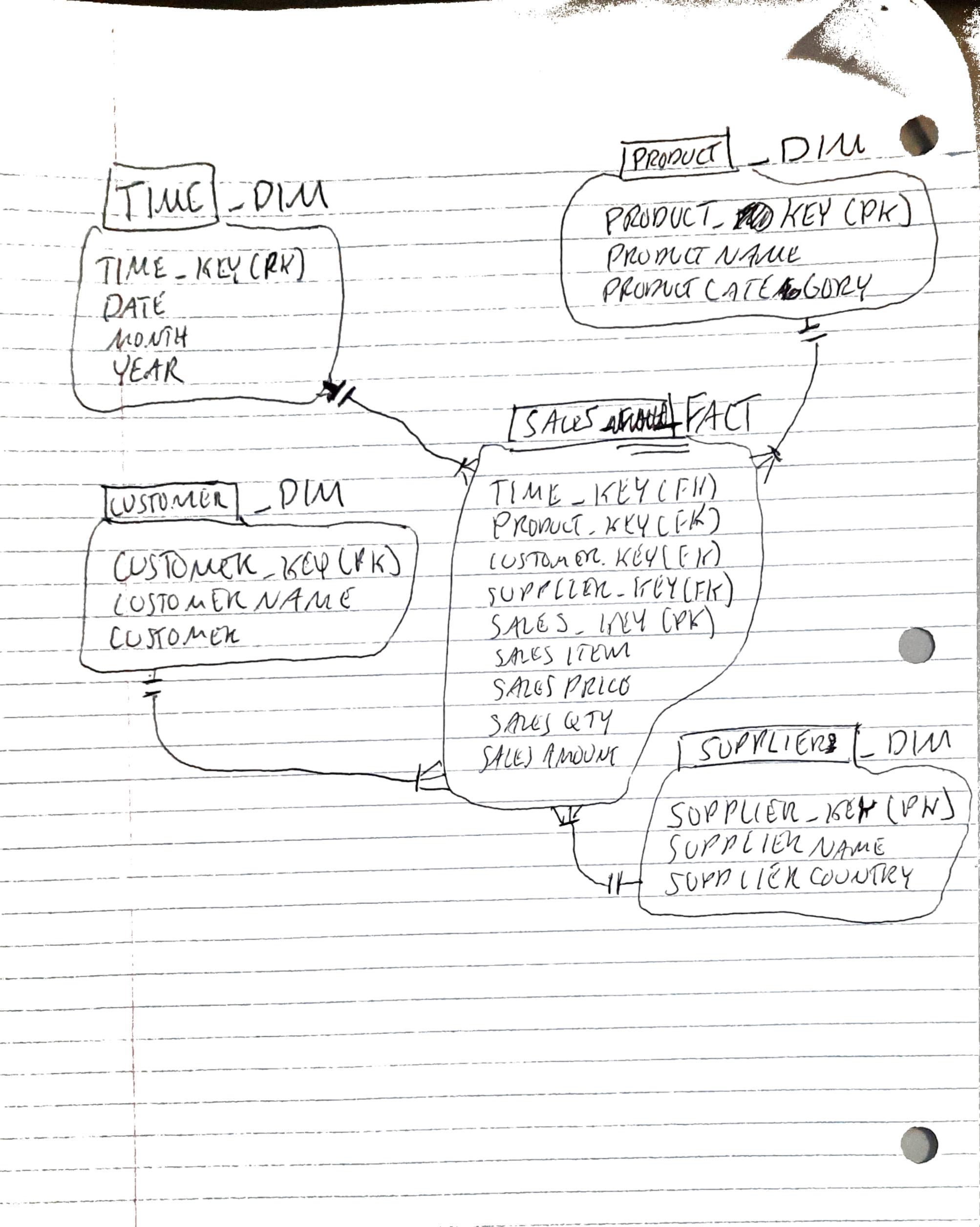
WHERE Salary = (SELECT MAX(Salary) FROM Employee)

**Section 6: 3 points**

Create a Retail Sales Company **Data Warehouse design** using **STAR schema** from following info. Make sure to indicate proper \_DIM and \_Fact tables and their PKs/FKs. You need to join those tables using JUST straight lines (optionality and cardinality relationships are NOT required).

Date, Month, Year, SupplierName, SupplierCountry, ProductName, ProductCategory, CustomerName, CustomerCity, SalesItem, SalesPrice, SalesQty, SalesAmount

Note: You may use any tool or just handwritten to create STAR schema data warehouse design.



☺ ☺ ☺ The END ☺ ☺ ☺