SMU SINGAPORE MANAGEMENT UNIVERSITY

IS112 DATA MANAGEMENT

Total marks: 25

SQL Sample Test

Student Name________________________ Student Email____________________________

INSTRUCTIONS TO STUDENTS

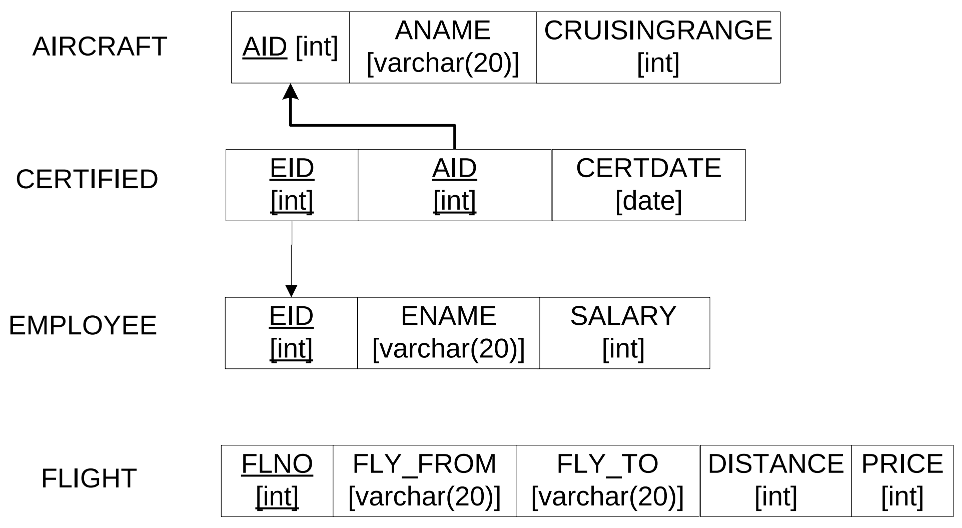
1. Total Number of pages is Six (6), including this cover page and two appendix (SQL Quick Reference) pages.
2. The final submission is one SQL file GX-Y.sql or GX-Y.txt, where X is the section (1-13) and Y is your user id (e.g. User bhzheng@smu.edu.sg has the file named as G1-bhzheng.sql)
3. State your NAME and STUDENT ID as first comment line; subsequently state the question number using comments followed by the answer. An example is given here. Notation # is for comments.

```
#Provided by Zheng Baihua (ID: 12345678)
#Qn.1
select * from .........
#Qn.2
select * from .........
```

   Note: You should provide only **one sql statement** as an answer to each question except for questions 1, 19 and 20.

4. **Return the test paper** to the instructor or TA before leaving the test venue.
5. **Log in to** the eLearn to submit the SQL file under **MySQL Test Submission**.
6. The test is 1 hour 50 minutes including time for online submission. Late submissions will have penalty.
7. This is a close book test, and you should NOT open any existing file in your computer. Any student who opens (or attempt to open) an existing file in his/her computer will be awarded **ZERO** for the test
8. The wireless connection should be disabled before the test, and students are allowed to enable the connection only when they do the submission. Any student who enables or attempts to enable the wireless connection during the test will be awarded **ZERO** for the test.

Following schema keeps data about Flight, Employee and Certification information. Note that the Employee relation describes pilots and other kinds of employee.



EMPLOYEE

EID	ENAME	SALARY
1	Jacob	85000
2	Michael	55000
3	Emily	80000
4	Ashley	110000
5	Daniel	80000
6	Olivia	70000

AIRCRAFT

AID	ANAME	CRUISINGRANGE
1	a1	800
2	a2b	700
3	a3	1000
4	a4b	1100
5	a5	1200

FLIGHT

FLNO	FLY_FROM	FLY_TO	DISTANCE	PRICE
1	LA	SF	600	65000
2	LA	SF	700	70000
3	LA	SF	800	90000
4	LA	NY	1000	85000
5	NY	LA	1100	95000

CERTIFIED

EID	AID	CERTDATE
1	1	2005-01-01
1	2	2001-01-01
1	3	2000-01-01
1	5	2000-01-01
2	3	2002-01-01
2	2	2003-01-01
3	3	2003-01-01
3	5	2004-01-01

Write the following queries in SQL. **No duplicates** should be printed in any of the answers (1-15:1.0 point each, 16-20: 2.0 point each).

1. Add a new column SPEAIRCRAFT (Type: INT) to the EMPLOYEE table which is a foreign key referring to AIRCRAFT'S AID. Then insert data as follows.

EID	ENAME	SALARY	SPEAIRCRAFT
1	Jacob	85000	1
2	Michael	55000	3
3	Emily	80000	4
4	Ashley	110000	3
5	Daniel	80000	NULL
6	Olivia	70000	NULL

2. Find the EID of those employees who are certified to fly Aircraft with AID equal 1 or CERTDATE falling in the year of 2003.
3. Print out the names of all the destinations in FLIGHT table according to the descending order of the names.
4. Print the EID of those employees with salary between 70000 and 100000 (inclusive).
5. Display the largest CRUISINGRANGE and the smallest CRUISINGRANGE of all the aircrafts.
6. Find the names and salaries of all the employees. If the employee is a pilot, print the names of all the aircrafts that he/she is certified to drive. (Note: an employee who is certified to drive at least one aircraft is considered as a pilot).
7. For each pair of (FLY_FROM, FLY_TO) in FLIGHT table, find the price of the cheapest flight between them.
8. Print the name and salary of every employee who is not a pilot and has salary higher than the average salary of those pilots who are certified to fly aircrafts with cruisingrange longer than 1000 (Note: an employee who is certified to drive at least one aircraft is considered as a pilot).
9. For each pilot who is certified to drive at least two aircrafts, print the EID, ENAME and the minimum CRUISINGRANGE of the aircrafts for which she/ he is certified.
10. Find the pilots who are certified to drive any aircraft with its name containing character 'b' (e.g., a2b). For each of these pilots, list his/her name, the names of the aircrafts with their name containing 'b', and the corresponding certified date.

11. Find the names of employees with salary less than the price of the cheapest route from LA to SF.
12. Find the names of aircrafts that are certified by Jacob **ONLY**.
13. Find the names of aircrafts such that **ALL** pilots certified to operate them have salaries between $60000 and $85000.
14. Print the names and salary of employee who has salary>70000 and are certified **ONLY** on aircrafts with cruising range longer than the distance of the flight with FLNO = 3.
15. Print the ENAMES of pilots who can operate planes with CRUISINGRANGE greater than 1000 miles but are not certified on any aircraft with name containing 'b' (e.g., a2b).
16. Find the year that issues most certificates (assumption: each record in CERTIFIED table corresponds to a certificate).
17. For each flight in FLIGHT table, find the name of the pilot who can fly that flight with lowest salary. Note: a pilot can fly the flight only when the pilot is certified to operate some aircraft with CRUISINGRANGE >= the distance of the flight.
18. Given a flight, the aircraft that has its CRUSINGRANGE >= distance of the flight and having the smallest (CRUISINGRANGE - the distance of the flight) is considered to be the most economical choice. For each flight in the FLIGHT table, find the most economical aircraft, together with number of pilots who can fly that aircraft. The output is in the form of (FLNO, AID, NUMBER).
19. Write a trigger to increase the salary of the pilot by $200 for each new certification he receives which is for aircrafts with cruising range of 1200 and above.
20. Write a stored procedure named 'sp_AircraftPilots' that takes in Aircraft AID as the input and displays the EID and ENAME of the lowest salaried pilots who are certified to fly the given aircraft. Note: there could be more than one pilot with the same lowest salary amount that can fly the aircraft. In addition, return the total number of pilots with the out variable '*total*'.

APPENDIX - SQL Quick Reference

Note: keywords are shown in uppercase.
Words appearing in italics are meant to be replaced.
Words or phrases shown within [ ] mean that they are optional.
… is indicated to mean that there could be more (of what is preceding …)
| is indicated to mean either – or

CREATE TABLE *table_name* (
column1 datatype [NOT NULL] [PRIMARY KEY],
column2 datatype ,
… ,
[CONSTRAINT *constraint_name* PRIMARY KEY (*column names)*],
[CONSTRAINT *constraint_name* FOREIGN KEY (*column names) REFERENCES table_name(column_names*)]
);

ALTER TABLE *table_name* ADD *column_name datatype;*
ALTER TABLE *table_name* ADD CONSTRAINT *constraint_name* FOREIGN KEY (*column_names) REFERENCES table_name*(*column_names*) ;
;

INSERT INTO *table_name* VALUES(*column1*, *column2*, …);
UPDATE *table_name* SET *column_name* = value1 [WHERE *condition*];
DELETE FROM *table_name* [WHERE *condition*];

SELECT *colum1*, *column2*, … FROM *table_name* [WHERE *condition*];

GROUP BY *column_names* : groups rows based on the given column name(s)

HAVING : clause that appears with GROUP BY and is used to filter groups

SELECT *column*, … FROM *table1* INNER | OUTER JOIN *table2* ON *table1.column_name* = *table2.column_name* [AND *condition*]

DISTINCT: return distinct values

ORDER BY : to order the output in ascending order ASC or descending order DESC

LIKE: used in a WHERE clause to search for a pattern in a column, used with % and _

IN : clause allows to specify multiple values and used in WHERE clause.
IS NULL: clause used in where clause to compare NULL values

BETWEEN: used to select values within a given range

UNION: is used to combine the result set of 2 or more SELECT statements (only distinct values)

UNION ALL: is used to combine the result set of 2 or more SELECT statements (allows duplicate values)

MIN(): used to get the smallest of the selected column

MAX(): used to get the largest of the selected column

AVG(): used to get the average of a numeric column

SUM(): used to compute the sum of a numeric column

COUNT(): used to return the number of rows that matches the criteria given in ()

```
CREATE TRIGGER trigger_name {BEFORE | AFTER}
{INSERT | UPDATE | DELETE}
ON table_name FOR EACH ROW
BEGIN
trigger body
END;
```

DELIMITER $$: keyword used to change the delimiter from the default ; to $$

NEW, OLD: keywords used in trigger to indicate the new and old values that are affected by the trigger.

SIGNAL SQLSTATE '45000' SET *MESSAGE_TEXT* = 'message';
used in trigger to throw a user defined message, hence the state 45000.

```
CREATE PROCEDURE procedure_name ([parameter(s)])
BEGIN
procedure body
END;
```

CALL *procedure_name([values]); to call the procedure*

DECLARE *variable_name data_type*;

SET *variable_name = value*;

SET variable_name = (SELECT *statement*);

```
IF boolean_expression THEN
    statements
[ELSEIF boolean_expression THEN
   statements
ELSE
    statements]
END IF;
```

```
WHILE Boolean_expression DO
   statements
END WHILE;
```