

Name (Last, First): _____

GT Username (e.g. gburdell3): _____

GT ID (e.g. 900-----): _____

Signature: _____

CS4400 – Introduction to Database Systems

Exam #1 – Database Basic Concepts, Three-Schema Architecture & Structured Query Language (SQL) DML

Versions [A, H, P, T]

- By writing your name at the top of this test **you are certifying that this test is entirely your own work per the Georgia Institute of Technology Academic Honor Code.**
- The exam is ***closed book*, *closed notes***, and ***no computer/calculator***.
- Using extra sheets of completely blank paper for scratch work is acceptable.
- Please **write the version of the Test Booklet on your Answer Sheet immediately**. You will lose points if you leave the version blank or write in the incorrect version letter.
- You must **turn in the Answer Sheet and the complete Test Booklet** to receive credit for the exam.
- Please **circle the final/official answer clearly in the Test Booklet** for each problem.
- You may ask the proctors for clarification but you are ultimately responsible for the answer you write on the paper.
- You may separate the pages if that makes it easier to work on the problems, but it's your responsibility to ensure that you reassemble and submit all Test Booklet pages before you leave.
- There are thirty-three (33) questions, but **you will only be graded on the basis of thirty (30) questions**. If you get more than 30 questions correct, you will keep the "bonus points."

Pace yourself, trust your studies, and good luck!

Versions [A, H, P, T]

(1) Logical data independence refers to the ability to modify the underlying conceptual schema without having to modify the external views.

- (a) True
- (b) False

(2) The following things are commonly true about the state of the database:

- (a) It describes the data that is actually stored in the database at a specific point in time
- (b) It changes less frequently than the schema of the database
- (c) All of the above
- (d) None of the above

(3) The concept that most appropriately expresses the separation of the operations of the programs that we use from the specific ways that we structure and store the underlying data is known as program-data independence.

- (a) True
- (b) False

(4) Which of the following statements are true?

- (a) SQL queries without DISTINCT always include duplicate entries in the result set
- (b) The presence of duplicates in the result set of an SQL query depends on the commands used within the query along with the use of keywords like ALL and DISTINCT
- (c) SQL is based on set theory and always removes duplicate entries from the result set
- (d) All of the above
- (e) None of the above

(5) Some of the main advantages of using relational database systems include:

- (a) SQL allows users to fundamentally ignore the underlying (and deeply technical) storage details of their data and (instead) focus on the conceptual structure of their data.
- (b) Relational database management systems often use special databases (e.g., catalogs) to store information (meta-data) about the structure of the other databases.
- (c) The three-schema architectures of many database systems are intended to allow you to change the low-level physical data structures and mapping details without having to change the higher-level conceptual schema or external views.
- (d) All of the above
- (e) None of the above

(6) Relational databases are often accessed using SQL, which is an imperative (i.e., procedural) language that allows the user to specifically direct how the database searches the records in the tables and assembles the result set.

- (a) True
- (b) False

(7) The information contained in the clauses of a Structured Query Language (SQL) query are normally "evaluated" (i.e., accessed) in the following sequence when the query is being processed (ignore ORDER BY for this problem):

- (a) SELECT – FROM – WHERE – GROUP BY – HAVING
- (b) FROM – WHERE – HAVING – GROUP BY – SELECT
- (c) SELECT – WHERE – GROUP BY – HAVING – FROM
- (d) FROM – GROUP BY – WHERE – HAVING – SELECT
- (e) FROM – WHERE – GROUP BY – HAVING – SELECT

(8) Database management systems are software packages/systems that can be used to create and manage one or more databases.

- (a) True
- (b) False

Determine which of the following SQL queries are functionally correct for the given English requirements. A query is correct if (and only if) it would return the correct results for all valid database states – not just the current database state. Duplicate rows are OK unless explicitly forbidden by the query requirements.

(9) Display all of the airport information in the order of their country first (A-Z), and then with busier airports listed before "less busy" airports.

- (a) select * from airports order by country, traffic desc;
- (b) select * from airports order by country asc, traffic;
- (c) All of the above
- (d) None of the above

(10) Display all of the information for any flights (without duplicates) that depart from or arrive into the ATL airport.

- (a) select * from flights where depart_from = 'ATL' or arrive_at = 'ATL';
- (b) select * from flights where depart_from like '%ATL' or arrive_at like 'ATL%';
- (c) All of the above
- (d) None of the above

(11) Display all of the passenger information for those passengers carrying more than one bag.

- (a) select * from passengers where bags > 1;
- (b) select * from passengers where bags between 2 and 3;
- (c) All of the above
- (d) None of the above

(12) Display the total number of passengers on flight 52.

- (a) select count(*) from passengers where taking_flight = 52;
- (b) select sum(pid) from passengers where taking_flight in (52);
- (c) All of the above
- (d) None of the above

(13) Display the flight number and total number of bags for each flight that has at least one passenger.

- (a) select taking_flight, sum(bags) from passengers group by taking_flight;
- (b) select sum(bags) from passengers group by taking_flight;
- (c) All of the above
- (d) None of the above

(14) Display the list of flight identifiers without duplicates.

- (a) select distinct taking_flight from passengers;
- (b) (select taking_flight from passengers) union all (select taking_flight from passengers);
- (c) All of the above
- (d) None of the above

(15) Display all of the airport information for airports located in a country that begins with the word 'United'.

- (a) select * from airports where country like 'United%';

(b) select * from airports where country like '%United_%';

(c) All of the above

(d) None of the above

(16) Display the airline, flight number and duration of all flights departing from China.

(a) select airline, flight_num, duration from flights where depart_from in

(select apcode from airports where country = 'China');

(b) select airline, flight_num, duration from flights where arrive_at not in

(select apcode from airports where country = 'China');

(c) All of the above

(d) None of the above

(17) Display the durations of the longest and shortest flights.

(a) select max(duration) as 'longest', min(duration) as 'shortest' from flights;

(b) (select 'longest' as 'category', max(duration) as 'duration' from flights) union

(select 'shortest', min(duration) from flights);

(c) All of the above

(d) None of the above

(18) Display the total airport traffic for each country which has fewer than three airports.

(a) select country, sum(traffic) from airports group by country having count(*) < 3;

(b) select sum(traffic) from airports group by country having count(*) < 3;

(c) All of the above

(d) None of the above

(19) Display the airline and flight number for any flight that has at least one passenger.

(a) select airline, flight_num from flights where fid in (select taking_flight from passengers);

(b) select airline, flight_num from flights, passengers group by airline, flight_num

having count(*) >= 1;

(c) All of the above

(d) None of the above

(20) Display the total number of flights that cost more than \$700 for each airline, but only for airlines that have fewer than three flights.

(a) select airline, count(*) from flights where cost > 700 and airline in

(select airline from flights group by airline having count(*) < 3) group by airline;

(b) select airline, count(*) from flights where cost > 700 group by airline having count(*) < 3;

(c) All of the above

(d) None of the above

(21) Display all passenger information for passengers on a flight that departed from the United States.

(a) select * from passengers where taking_flight in (select fid from flights
where depart_from in (select apcode from airports where country = 'United States'));

(b) select * from passengers where taking_flight in (select fid from flights, airports
where depart_from = apcode or country = 'United States');

(c) All of the above

(d) None of the above

(22) Display the airport code and the number of flights arriving at that airport.

All airports must be included in your result set, and '0' or 'null' values will be interpreted as zero flights.

(a) select apcode, count(*) from airports left outer join flights

- on apcode = arrive_at group by apcode, apname;
(b) select apcode, count(*) from flights left outer join airports
on apcode = arrive_at group by apcode, apname;
(c) All of the above
(d) None of the above

Determine the number of rows that will be produced by the following SQL queries.

(23) Display the flight identifier, airline, and flight number along with the number of passengers on that flight.

All flights must be included in your result set, and '0' or 'null' values will be interpreted as zero passengers.

- (a) select fid, airline, flight_num, count(distinct pid) from flights left outer join passengers
on fid = taking_flight group by fid, airline, flight_num;
(b) select fid, airline, flight_num, count(*) from flights left outer join passengers
on fid = taking_flight group by fid, airline, flight_num;
(c) All of the above
(d) None of the above

(24) select airline, count(*) from flights group by airline;

- (a) 0
(b) 6
(c) 7
(d) 8
(e) None of the above (also includes errors in query)

(25) select airline, count(*) from flights where fid between 55 and 59 group by airline;

- (a) 0
(b) 3
(c) 4
(d) 5
(e) None of the above (also includes errors in query)

(26) select taking_flight, count(*), sum(bags) from passengers where pid between 2 and 6
group by taking_flight having count(*) < 2;

- (a) 0
(b) 2
(c) 3
(d) 4
(e) None of the above (also includes errors in query)

(27) select distinct depart_from from flights where airline like '%China%' or airline like '%Air%';

- (a) 0
(b) 4
(c) 5
(d) 6
(e) None of the above (also includes errors in query)

(28) (select airline from flights where depart_from = 'LAX') union (select airline from flights where
arrive_at = 'LAX');

- (a) 0
(b) 1

- (c) 2
- (d) 3
- (e) None of the above (also includes errors in query)

(29) select * from flights where depart_from in (select apcode from airports where country = 'United States')

and fid in (select taking_flight from passengers where bags < 2);

- (a) 0
- (b) 1
- (c) 2
- (d) 3
- (e) None of the above (also includes errors in query)

(30) select * from flights, passengers where bags < 1;

- (a) 0
- (b) 60
- (c) 70
- (d) 90
- (e) None of the above (also includes errors in query)

(31) select * from flights join passengers on fid = taking_flight;

- (a) 0
- (b) 5
- (c) 10
- (d) 20
- (e) None of the above (also includes errors in query)

(32) select * from flights left outer join passengers on fid = taking_flight;

- (a) 0
- (b) 5
- (c) 10
- (d) 20
- (e) None of the above (also includes errors in query)

(33) select * from airports, flights where country in ('United States', 'United Arab Emirates') and (apcode = depart_from or apcode = arrive_at);

- (a) 0
- (b) 7
- (c) 9
- (d) 11
- (e) None of the above (also includes errors in query)

Any remaining space is yours for scratch work.

Use the following tables for all of the SQL related questions. These tables are part of a database about air travel. The air travel information below is all that you need to complete the questions.

This table is named **airports**, and each record contains facts about: a unique identifier for that airport (*apcode*); the name of the airport (*apname*), along with its location (i.e., *country*, *state*, and *city*); and, a relative measure of how busy the airport is in terms of passenger flow (*traffic*).

apcode	apname	country	state	city	traffic
ATL	Hartsfield Jackson	United States	Georgia	Atlanta	54.4
DXB	Dubai International	United Arab Emirates	Dubai	Garhoud	41.3
HND	Tokyo Haneda	Japan	Tokyo	Ota	41.4
LAX	Los Angeles	United States	California	Los Angeles	43.0
LHR	London Heathrow	United Kingdom	London	Hillingdon	38.8
ORD	Chicago OHare	United States	Illinois	Chicago	39.5
PEK	Beijing Capital	China	Beijing	Chaoyang-Shunyi	49.2
PVG	Shanghai Pudong	China	Shanghai	Pudong	38.2

This table is named **flights**, and each record contains facts about: a unique identifier for that flight (*fid*); the *airline* company and flight number (*flight_num*); the airports of departure (*depart_from*) and arrival (*arrive_at*); the *duration* of the flight including layovers; and, the *cost* of the flight.

fid	airline	flight_num	depart_from	arrive_at	duration	cost
50	Air China	7450	ATL	PEK	19	1051.00
51	United	7983	LAX	HND	13	913.00
52	Turkish Airlines	763_5	DXB	ORD	20	1296.00
53	Swiss Air	317_188	LHR	PVG	15	533.00
54	Delta	1735	ATL	LAX	5	604.00
55	United	660	LAX	ORD	4	451.00
56	China Eastern	271	PEK	PVG	2	468.00
57	United	1027	ORD	ATL	2	340.00
58	Emirates	303	PVG	DXB	11	877.00
59	Emirates	3	DXB	LHR	8	709.00

This table is named **passengers**, and each record contains facts about: a unique identifier for that passenger (*pid*); the first (*fname*) and last (*lname*) names of the passenger; the identifier of the flight that they are taking (*taking_flight*); and, the number of *bags* they are carrying.

pid	fname	lname	taking_flight	bags
1	Susanna	Delacruz	51	1
2	Bryony	Stark	52	3
3	Michael	Kayden	52	2
4	Kamila	Ferry	52	3
5	Abigayle	Velez	55	3
6	Laurence	McCormick	56	0
7	Willow	McDaniel	56	1
8	Portia	Banks	58	1
9	Samuel	Carpenter	59	0

Some key facts about the data, including how the tables are related:

- The *taking_flight* identifier values in the **passengers** table must represent a valid flight identifier (*fid*) from the **flights** table.
- The airport identifier values (*depart_from* and *arrive_at*) in the **flights** table must represent valid airport identifiers (*apcode*) from the **airports** table

Name (Last, First): _____

GT Username (e.g. gburdell3): _____

GT ID (e.g. 900-----): _____

Signature: _____

CS4400 – Introduction to Database Systems

Exam #1 – Database Basic Concepts, Three-Schema Architecture & Structured Query Language (SQL) DML

Versions [B, J, W, R, X]

- By writing your name at the top of this test **you are certifying that this test is entirely your own work per the Georgia Institute of Technology Academic Honor Code.**
- The exam is ***closed book, closed notes***, and ***no computer/calculator***.
- Using extra sheets of completely blank paper for scratch work is acceptable.
- Please **write the version of the Test Booklet on your Answer Sheet immediately**. You will lose points if you leave the version blank or write in the incorrect version letter.
- You must **turn in the Answer Sheet and the complete Test Booklet** to receive credit for the exam.
- Please **circle the final/official answer clearly in the Test Booklet** for each problem.
- You may ask the proctors for clarification but you are ultimately responsible for the answer you write on the paper.
- You may separate the pages if that makes it easier to work on the problems, but it's your responsibility to ensure that you reassemble and submit all Test Booklet pages before you leave.
- There are thirty-three (33) questions, but **you will only be graded on the basis of thirty (30) questions**. If you get more than 30 questions correct, you will keep the "bonus points."

Pace yourself, trust your studies, and good luck!

Versions [B, J, W, R, X]

(1) Logical data independence refers to the ability to modify the underlying conceptual schema without having to modify the external views.

- (a) True
- (b) False

(2) The following things are commonly true about the state of the database:

- (a) It describes the design of the database schema at a specific point in time
- (b) It changes more frequently than the schema of the database
- (c) All of the above
- (d) None of the above

(3) The concept that most appropriately expresses the separation of the operations of the programs that we use from the specific ways that we structure and store the underlying data is known as program-metadata compilation.

- (a) True
- (b) False

(4) Which of the following statements are true?

- (a) The presence of duplicates in the result set of an SQL query depends on the commands used within the query along with the use of keywords like ALL and DISTINCT
- (b) SQL queries without DISTINCT always include duplicate entries in the result set
- (c) SQL is based on set theory and always removes duplicate entries from the result set
- (d) All of the above
- (e) None of the above

(5) Some of the main advantages of using relational database systems include:

- (a) SQL allows users to fundamentally ignore the underlying (and deeply technical) storage details of their data and (instead) focus on the conceptual structure of their data.
- (b) Relational database management systems often use special databases (e.g., catalogs) to store information (meta-data) about the structure of the other databases.
- (c) The three-schema architectures of many database systems are intended to allow you to change the low-level physical data structures and mapping details without having to change the higher-level conceptual schema or external views.
- (d) All of the above
- (e) None of the above

(6) Relational databases are often accessed using SQL, which is an imperative (i.e., procedural) language that allows the user to specifically direct how the database searches the records in the tables and assembles the result set.

- (a) True
- (b) False

(7) The information contained in the clauses of a Structured Query Language (SQL) query are normally "evaluated" (i.e., accessed) in the following sequence when the query is being processed (ignore ORDER BY for this problem):

- (a) SELECT – FROM – WHERE – GROUP BY – HAVING
- (b) FROM – WHERE – HAVING – GROUP BY – SELECT
- (c) SELECT – WHERE – GROUP BY – HAVING – FROM
- (d) FROM – GROUP BY – WHERE – HAVING – SELECT
- (e) FROM – WHERE – GROUP BY – HAVING – SELECT

(8) Databases are software packages/systems that can be used to create and manage one or more databases management systems.

- (a) True
- (b) False

Determine which of the following SQL queries are functionally correct for the given English requirements. A query is correct if (and only if) it would return the correct results for all valid database states – not just the current database state. Duplicate rows are OK unless explicitly forbidden by the query requirements.

(9) Display all of the airport information in the order of their country first (A-Z), and then with busier airports listed before "less busy" airports.

- (a) select * from airports order by country, traffic desc;
- (b) select * from airports order by country asc, traffic desc;
- (c) All of the above
- (d) None of the above

(10) Display all of the information for any flights (without duplicates) that depart from or arrive into the ATL airport.

- (a) select * from flights where depart_from like '%ATL' or arrive_at like 'ATL%';
- (b) (select * from flights where arrive_at = 'ATL') union all (select * from flights where depart_from = 'ATL');
- (c) All of the above
- (d) None of the above

(11) Display all of the passenger information for those passengers carrying more than one bag.

- (a) select * from passengers where bags between 2 and 3;
- (b) select count(*) from passengers where bags > 1;
- (c) All of the above
- (d) None of the above

(12) Display the total number of passengers on flight 52.

- (a) select count(*) from passengers where taking_flight like '%52%';
- (b) select count(distinct pid) from passengers where taking_flight = 52;
- (c) All of the above
- (d) None of the above

(13) Display the flight number and total number of bags for each flight that has at least one passenger.

- (a) select distinct taking_flight from passengers;
- (b) select sum(bags) from passengers group by taking_flight;
- (c) All of the above
- (d) None of the above

(14) Display the list of flight identifiers without duplicates.

- (a) select taking_flight from passengers;
- (b) (select taking_flight from passengers) union (select taking_flight from passengers);
- (c) All of the above
- (d) None of the above

(15) Display all of the airport information for airports located in a country that begins with the word 'United'.

- (a) select * from airports where country like 'United%';
- (b) select * from airports where country like '%United_%';
- (c) All of the above
- (d) None of the above

(16) Display the airline, flight number and duration of all flights departing from China.

- (a) select airline, flight_num, duration from flights where depart_from in (select apcode from airports where country = 'China');
- (b) select airline, flight_num, duration from flights where arrive_at not in (select apcode from airports where country = 'China');
- (c) All of the above
- (d) None of the above

(17) Display the durations of the longest and shortest flights.

- (a) select max(duration) as 'longest', min(duration) as 'shortest' from flights;
- (b) (select 'longest' as 'category', max(duration) as 'duration' from flights) union (select 'shortest', min(duration) from flights);
- (c) All of the above
- (d) None of the above

(18) Display the total airport traffic for each country which has fewer than three airports.

- (a) select country, count(traffic) from airports group by country having count(*) < 3;
- (b) select sum(traffic) from airports group by country having count(*) < 3;
- (c) All of the above
- (d) None of the above

(19) Display the airline and flight number for any flight that has at least one passenger.

- (a) select airline, flight_num from flights where fid in (select taking_flight from passengers);
- (b) select airline, flight_num from flights, passengers group by airline, flight_num having count(*) >= 1;
- (c) All of the above
- (d) None of the above

(20) Display the total number of flights that cost more than \$700 for each airline, but only for airlines that have fewer than three flights.

- (a) select airline, count(*) from flights where cost > 700 group by airline having count(*) < 3;
- (b) select airline, count(*) from flights where cost > 700 and airline in (select airline from flights group by airline having count(*) < 3) group by airline;
- (c) All of the above
- (d) None of the above

(21) Display all passenger information for passengers on a flight that departed from the United States.

- (a) select * from passengers where taking_flight in (select fid from flights, airports where depart_from = apcode and country = 'United States');
- (b) select * from passengers where taking_flight in (select fid from flights where depart_from in (select apcode from airports where country = 'United States'));
- (c) All of the above
- (d) None of the above

(22) Display the airport code and the number of flights arriving at that airport.

All airports must be included in your result set, and '0' or 'null' values will be interpreted as zero flights.

- (a) select apcode, total_in from airports,
 (select arrive_at, count(*) as 'total_in' from flights group by arrive_at) as temp
 where apcode = arrive_at;
 (b) (select apcode, total_in from airports,
 (select arrive_at, count(*) as 'total_in' from flights group by arrive_at) as temp
 where apcode = arrive_at) union
 (select apcode, 0 from airports where apcode not in (select arrive_at from flights));
 (c) All of the above
 (d) None of the above

Determine the number of rows that will be produced by the following SQL queries.

(23) Display the flight identifier, airline, and flight number along with the number of passengers on that flight.

All flights must be included in your result set, and '0' or 'null' values will be interpreted as zero passengers.

- (a) select fid, airline, flight_num, count(distinct pid) from flights left outer join passengers
 on fid = taking_flight group by fid, airline, flight_num;
 (b) select fid, airline, flight_num, count(*) from flights left outer join passengers
 on fid = taking_flight group by fid, airline, flight_num;
 (c) All of the above
 (d) None of the above

(24) select airline, depart_from, count(*) from flights group by airline, depart_from;

- (a) 0
 (b) 6
 (c) 7
 (d) 8
 (e) None of the above (also includes errors in query)

(25) select depart_from, count(*) from flights where fid between 50 and 54 group by depart_from;

- (a) 0
 (b) 3
 (c) 4
 (d) 5
 (e) None of the above (also includes errors in query)

(26) select taking_flight, count(*), sum(bags) from passengers where pid between 2 and 6
 group by taking_flight having count(*) < 2;

- (a) 0
 (b) 2
 (c) 3
 (d) 4
 (e) None of the above (also includes errors in query)

(27) select distinct depart_from from flights where airline not like '%China%' and airline not like '%Air%';

- (a) 0
 (b) 4
 (c) 5
 (d) 6
 (e) None of the above (also includes errors in query)

(28) (select airline from flights where depart_from = 'ATL') union (select airline from flights where arrive_at = 'ATL');

- (a) 0
- (b) 1
- (c) 2
- (d) 3
- (e) None of the above (also includes errors in query)

(29) select * from flights where arrive_at in (select apcode from airports where country = 'United States')

and fid in (select taking_flight from passengers where bags > 1);

- (a) 0
- (b) 1
- (c) 2
- (d) 3
- (e) None of the above (also includes errors in query)

(30) select * from airports, flights where country <> 'China';

- (a) 0
- (b) 60
- (c) 70
- (d) 80
- (e) None of the above (also includes errors in query)

(31) select * from flights join passengers on duration = pid;

- (a) 0
- (b) 5
- (c) 10
- (d) 20
- (e) None of the above (also includes errors in query)

(32) select * from airports left outer join flights on apcode = depart_from;

- (a) 0
- (b) 5
- (c) 10
- (d) 20
- (e) None of the above (also includes errors in query)

(33) select * from airports, flights where country in ('United States', 'United Arab Emirates') and (apcode = depart_from or apcode = arrive_at);

- (a) 0
- (b) 7
- (c) 9
- (d) 11
- (e) None of the above (also includes errors in query)

Any remaining space is yours for scratch work.

Use the following tables for all of the SQL related questions. These tables are part of a database about air travel. The air travel information below is all that you need to complete the questions.

This table is named **airports**, and each record contains facts about: a unique identifier for that airport (*apcode*); the name of the airport (*apname*), along with its location (i.e., *country*, *state*, and *city*); and, a relative measure of how busy the airport is in terms of passenger flow (*traffic*).

apcode	apname	country	state	city	traffic
ATL	Hartsfield Jackson	United States	Georgia	Atlanta	54.4
DXB	Dubai International	United Arab Emirates	Dubai	Garhoud	41.3
HND	Tokyo Haneda	Japan	Tokyo	Ota	41.4
LAX	Los Angeles	United States	California	Los Angeles	43.0
LHR	London Heathrow	United Kingdom	London	Hillingdon	38.8
ORD	Chicago OHare	United States	Illinois	Chicago	39.5
PEK	Beijing Capital	China	Beijing	Chaoyang-Shunyi	49.2
PVG	Shanghai Pudong	China	Shanghai	Pudong	38.2

This table is named **flights**, and each record contains facts about: a unique identifier for that flight (*fid*); the *airline* company and flight number (*flight_num*); the airports of departure (*depart_from*) and arrival (*arrive_at*); the *duration* of the flight including layovers; and, the *cost* of the flight.

fid	airline	flight_num	depart_from	arrive_at	duration	cost
50	Air China	7450	ATL	PEK	19	1051.00
51	United	7983	LAX	HND	13	913.00
52	Turkish Airlines	763_5	DXB	ORD	20	1296.00
53	Swiss Air	317_188	LHR	PVG	15	533.00
54	Delta	1735	ATL	LAX	5	604.00
55	United	660	LAX	ORD	4	451.00
56	China Eastern	271	PEK	PVG	2	468.00
57	United	1027	ORD	ATL	2	340.00
58	Emirates	303	PVG	DXB	11	877.00
59	Emirates	3	DXB	LHR	8	709.00

This table is named **passengers**, and each record contains facts about: a unique identifier for that passenger (*pid*); the first (*fname*) and last (*lname*) names of the passenger; the identifier of the flight that they are taking (*taking_flight*); and, the number of *bags* they are carrying.

pid	fname	lname	taking_flight	bags
1	Susanna	Delacruz	51	1
2	Bryony	Stark	52	3
3	Michael	Kayden	52	2
4	Kamila	Ferry	52	3
5	Abigayle	Velez	55	3
6	Laurence	McCormick	56	0
7	Willow	McDaniel	56	1
8	Portia	Banks	58	1
9	Samuel	Carpenter	59	0

Some key facts about the data, including how the tables are related:

- The *taking_flight* identifier values in the **passengers** table must represent a valid flight identifier (*fid*) from the **flights** table.
- The airport identifier values (*depart_from* and *arrive_at*) in the **flights** table must represent valid airport identifiers (*apcode*) from the **airports** table.

Name (Last, First): _____

GT Username (e.g. gburdell3): _____

GT ID (e.g. 900-----): _____

Signature: _____

CS4400 – Introduction to Database Systems

Exam #1 – Database Basic Concepts, Three-Schema Architecture & Structured Query Language (SQL) DML

Versions [C, F, N, Q, Y]

- By writing your name at the top of this test **you are certifying that this test is entirely your own work per the Georgia Institute of Technology Academic Honor Code.**
- The exam is ***closed book*, *closed notes***, and ***no computer/calculator***.
- Using extra sheets of completely blank paper for scratch work is acceptable.
- Please **write the version of the Test Booklet on your Answer Sheet immediately**. You will lose points if you leave the version blank or write in the incorrect version letter.
- You must **turn in the Answer Sheet and the complete Test Booklet** to receive credit for the exam.
- Please **circle the final/official answer clearly in the Test Booklet** for each problem.
- You may ask the proctors for clarification but you are ultimately responsible for the answer you write on the paper.
- You may separate the pages if that makes it easier to work on the problems, but it's your responsibility to ensure that you reassemble and submit all Test Booklet pages before you leave.
- There are thirty-three (33) questions, but **you will only be graded on the basis of thirty (30) questions**. If you get more than 30 questions correct, you will keep the "bonus points."

Pace yourself, trust your studies, and good luck!

Versions [C, F, N, Q, Y]

(1) Physical data independence refers to the ability to modify the underlying database file system structures, indexes and access methods without having to modify the conceptual schema.

- (a) True
- (b) False

(2) The following things are commonly true about the schema of the database:

- (a) It describes the structure of the database at a specific point in time
- (b) It changes less frequently than the state of the database
- (c) All of the above
- (d) None of the above

(3) The concept that most appropriately expresses the separation of the operations of the programs that we use from the specific ways that we structure and store the underlying data is known as program-metadata compilation.

- (a) True
- (b) False

(4) Which of the following statements are true?

- (a) The presence of duplicates in the result set of an SQL query depends on the commands used within the query along with the use of keywords like ALL and DISTINCT
- (b) SQL queries without DISTINCT always include duplicate entries in the result set
- (c) SQL is based on set theory and always removes duplicate entries from the result set
- (d) All of the above
- (e) None of the above

(5) Some of the main advantages of using relational database systems include:

- (a) SQL requires users to understand the underlying (and deeply technical) storage details of their data in order to be able to query the data.
- (b) Relational database management systems require you to create structures like tables and views that cannot be changed once they have been created.
- (c) The three-schema architectures of many database systems are intended to allow you to change the low-level physical data structures and mapping details without having to change the higher-level conceptual schema or external views.
- (d) All of the above
- (e) None of the above

(6) Relational databases are often accessed using SQL, which is a declarative language that allows the user to specify the desired result set without having to direct how the database searches the tables.

- (a) True
- (b) False

(7) The information contained in the clauses of a Structured Query Language (SQL) query are normally "evaluated" (i.e., accessed) in the following sequence when the query is being processed (ignore ORDER BY for this problem):

- (a) SELECT – FROM – WHERE – GROUP BY – HAVING
- (b) FROM – GROUP BY – WHERE – HAVING – SELECT
- (c) FROM – WHERE – GROUP BY – HAVING – SELECT
- (d) SELECT – WHERE – GROUP BY – HAVING – FROM
- (e) FROM – WHERE – HAVING – GROUP BY – SELECT

(8) Databases are software packages/systems that can be used to create and manage one or more databases management systems.

- (a) True
- (b) False

Determine which of the following SQL queries are functionally correct for the given English requirements. A query is correct if (and only if) it would return the correct results for all valid database states – not just the current database state. Duplicate rows are OK unless explicitly forbidden by the query requirements.

(9) Display all of the airport information in the order of their country first (A-Z), and then with busier airports listed before "less busy" airports.

- (a) select * from airports order by country, traffic desc;
- (b) select * from airports order by country asc, traffic desc;
- (c) All of the above
- (d) None of the above

(10) Display all of the information for any flights (without duplicates) that depart from or arrive into the ATL airport.

- (a) (select * from flights where arrive_at = 'ATL') union all (select * from flights where depart_from = 'ATL');
- (b) select * from flights where depart_from like 'ATL' or arrive_at like 'ATL';
- (c) All of the above
- (d) None of the above

(11) Display all of the passenger information for those passengers carrying more than one bag.

- (a) select * from passengers where bags > 1;
- (b) select * from passengers where bags between 2 and 3;
- (c) All of the above
- (d) None of the above

(12) Display the total number of passengers on flight 52.

- (a) select count(*) from passengers where taking_flight in (52);
- (b) select count(distinct pid) from passengers where taking_flight = 52;
- (c) All of the above
- (d) None of the above

(13) Display the flight number and total number of bags for each flight that has at least one passenger.

- (a) select taking_flight, count(bags) from passengers group by taking_flight;
- (b) select taking_flight, sum(bags) from passengers group by taking_flight having count(*) >= 1;
- (c) All of the above
- (d) None of the above

(14) Display the list of flight identifiers without duplicates.

- (a) select distinct taking_flight from passengers;
- (b) (select taking_flight from passengers) union (select taking_flight from passengers);
- (c) All of the above
- (d) None of the above

(15) Display all of the airport information for airports located in a country that begins with the word 'United'.

- (a) select * from airports where country like 'United_';
- (b) select * from airports where country like '%United_%';
- (c) All of the above
- (d) None of the above

(16) Display the airline, flight number and duration of all flights departing from China.

- (a) select airline, flight_num, duration from flights where depart_from in ('PEK', 'PVG');
- (b) select airline, flight_num, duration from flights join airports on depart_from = apcode where country = 'China';
- (c) All of the above
- (d) None of the above

(17) Display the durations of the longest and shortest flights.

- (a) select max(duration) as 'longest', min(duration) as 'shortest' from flights;
- (b) (select 'longest' as 'category', max(duration) as 'duration' from flights) union (select 'shortest', min(duration) from flights);
- (c) All of the above
- (d) None of the above

(18) Display the total airport traffic for each country which has fewer than three airports.

- (a) select country, sum(traffic) from airports where country in (select country from airports group by country having count(*) < 3) group by country;
- (b) select country, sum(traffic) from airports group by country having count(*) < 3;
- (c) All of the above
- (d) None of the above

(19) Display the airline and flight number for any flight that has at least one passenger.

- (a) select airline, flight_num from flights, passengers where fid = taking_flight group by airline, flight_num having count(*) > 1;
- (b) select airline, flight_num from flights join passengers on fid = taking_flight group by airline, flight_num having count(*) >= 1;
- (c) All of the above
- (d) None of the above

(20) Display the total number of flights that cost more than \$700 for each airline, but only for airlines that have fewer than three flights.

- (a) select airline, count(*) from flights where cost > 700 and airline in (select airline from flights group by airline having count(*) < 3) group by airline;
- (b) select airline, count(*) from flights where cost > 700 group by airline having count(*) < 3;
- (c) All of the above
- (d) None of the above

(21) Display all passenger information for passengers on a flight that departed from the United States.

- (a) select * from passengers where taking_flight in (select fid from flights where depart_from in (select apcode from airports where country = 'United States'));
- (b) select * from passengers where taking_flight in (select fid from flights, airports where depart_from = apcode or country = 'United States');
- (c) All of the above
- (d) None of the above

(22) Display the airport code and the number of flights arriving at that airport.

All airports must be included in your result set, and '0' or 'null' values will be interpreted as zero flights.

- (a) select apcode, total_in from airports,
(select arrive_at, count(*) as 'total_in' from flights group by arrive_at) as temp
where apcode = arrive_at;
- (b) select apcode, count(*) from flights left outer join airports
on apcode = arrive_at group by apcode, apname;
- (c) All of the above
- (d) None of the above

Determine the number of rows that will be produced by the following SQL queries.

(23) Display the flight identifier, airline, and flight number along with the number of passengers on that flight.

All flights must be included in your result set, and '0' or 'null' values will be interpreted as zero passengers.

- (a) select fid, airline, flight_num, total_riders from flights,
(select taking_flight, count(*) as 'total_riders' from passengers group by taking_flight) as temp
where fid = taking_flight;
- (b) select fid, airline, flight_num, count(*) from flights left outer join passengers
on fid = taking_flight group by fid, airline, flight_num;
- (c) All of the above
- (d) None of the above

(24) select arrive_at, count(*) from flights group by arrive_at;

- (a) 0
- (b) 6
- (c) 7
- (d) 8
- (e) None of the above (also includes errors in query)

(25) select arrive_at, count(*) from flights where fid between 52 and 56 group by arrive_at;

- (a) 0
- (b) 3
- (c) 4
- (d) 5
- (e) None of the above (also includes errors in query)

(26) select taking_flight, count(*), sum(bags) from passengers where pid between 1 and 6
group by taking_flight having count(*) < 2;

- (a) 0
- (b) 2
- (c) 3
- (d) 4
- (e) None of the above (also includes errors in query)

(27) select distinct arrive_at from flights where airline like '%China%' or airline like '%Air%';

- (a) 0
- (b) 4
- (c) 5
- (d) 6
- (e) None of the above (also includes errors in query)

(28) (select airline from flights where depart_from = 'LAX') union (select airline from flights where arrive_at = 'LAX');

- (a) 0
- (b) 1
- (c) 2
- (d) 3
- (e) None of the above (also includes errors in query)

(29) select * from flights where depart_from in (select apcode from airports where country = 'China') and fid in (select taking_flight from passengers where bags > 1);

- (a) 0
- (b) 1
- (c) 2
- (d) 3
- (e) None of the above (also includes errors in query)

(30) select * from airports, flights where country <> 'United States';

- (a) 0
- (b) 60
- (c) 70
- (d) 80
- (e) None of the above (also includes errors in query)

(31) select * from airports join flights on apcode = arrive_at;

- (a) 0
- (b) 5
- (c) 10
- (d) 20
- (e) None of the above (also includes errors in query)

(32) select * from flights left outer join passengers on duration = pid;

- (a) 0
- (b) 5
- (c) 10
- (d) 20
- (e) None of the above (also includes errors in query)

(33) select * from airports, flights where country in ('China', 'United Kingdom') and (apcode = depart_from or apcode = arrive_at);

- (a) 0
- (b) 7
- (c) 9
- (d) 11
- (e) None of the above (also includes errors in query)

Any remaining space is yours for scratch work.

Use the following tables for all of the SQL related questions. These tables are part of a database about air travel. The air travel information below is all that you need to complete the questions.

This table is named **airports**, and each record contains facts about: a unique identifier for that airport (*apcode*); the name of the airport (*apname*), along with its location (i.e., *country*, *state*, and *city*); and, a relative measure of how busy the airport is in terms of passenger flow (*traffic*).

apcode	apname	country	state	city	traffic
ATL	Hartsfield Jackson	United States	Georgia	Atlanta	54.4
DXB	Dubai International	United Arab Emirates	Dubai	Garhoud	41.3
HND	Tokyo Haneda	Japan	Tokyo	Ota	41.4
LAX	Los Angeles	United States	California	Los Angeles	43.0
LHR	London Heathrow	United Kingdom	London	Hillingdon	38.8
ORD	Chicago OHare	United States	Illinois	Chicago	39.5
PEK	Beijing Capital	China	Beijing	Chaoyang-Shunyi	49.2
PVG	Shanghai Pudong	China	Shanghai	Pudong	38.2

This table is named **flights**, and each record contains facts about: a unique identifier for that flight (*fid*); the *airline* company and flight number (*flight_num*); the airports of departure (*depart_from*) and arrival (*arrive_at*); the *duration* of the flight including layovers; and, the *cost* of the flight.

fid	airline	flight_num	depart_from	arrive_at	duration	cost
50	Air China	7450	ATL	PEK	19	1051.00
51	United	7983	LAX	HND	13	913.00
52	Turkish Airlines	763_5	DXB	ORD	20	1296.00
53	Swiss Air	317_188	LHR	PVG	15	533.00
54	Delta	1735	ATL	LAX	5	604.00
55	United	660	LAX	ORD	4	451.00
56	China Eastern	271	PEK	PVG	2	468.00
57	United	1027	ORD	ATL	2	340.00
58	Emirates	303	PVG	DXB	11	877.00
59	Emirates	3	DXB	LHR	8	709.00

This table is named **passengers**, and each record contains facts about: a unique identifier for that passenger (*pid*); the first (*fname*) and last (*lname*) names of the passenger; the identifier of the flight that they are taking (*taking_flight*); and, the number of *bags* they are carrying.

pid	fname	lname	taking_flight	bags
1	Susanna	Delacruz	51	1
2	Bryony	Stark	52	3
3	Michael	Kayden	52	2
4	Kamila	Ferry	52	3
5	Abigayle	Velez	55	3
6	Laurence	McCormick	56	0
7	Willow	McDaniel	56	1
8	Portia	Banks	58	1
9	Samuel	Carpenter	59	0

Some key facts about the data, including how the tables are related:

- The *taking_flight* identifier values in the **passengers** table must represent a valid flight identifier (*fid*) from the **flights** table.

The airport identifier values (*depart_from* and *arrive_at*) in the **flights** table must represent valid airport identifiers (*apcode*) from the **airports** table.

Name (Last, First): _____

GT Username (e.g. gburdell3): _____

GT ID (e.g. 900-----): _____

Signature: _____

CS4400 – Introduction to Database Systems

Exam #1 – Database Basic Concepts, Three-Schema Architecture & Structured Query Language (SQL) DML

Versions [D, K, M, S]

- By writing your name at the top of this test **you are certifying that this test is entirely your own work per the Georgia Institute of Technology Academic Honor Code.**
- The exam is ***closed book***, ***closed notes***, and ***no computer/calculator***.
- Using extra sheets of completely blank paper for scratch work is acceptable.
- Please **write the version of the Test Booklet on your Answer Sheet immediately**. You will lose points if you leave the version blank or write in the incorrect version letter.
- You must **turn in the Answer Sheet and the complete Test Booklet** to receive credit for the exam.
- Please **circle the final/official answer clearly in the Test Booklet** for each problem.
- You may ask the proctors for clarification but you are ultimately responsible for the answer you write on the paper.
- You may separate the pages if that makes it easier to work on the problems, but it's your responsibility to ensure that you reassemble and submit all Test Booklet pages before you leave.
- There are thirty-three (33) questions, but **you will only be graded on the basis of thirty (30) questions**. If you get more than 30 questions correct, you will keep the "bonus points."

Pace yourself, trust your studies, and good luck!

Versions [D, K, M, S]

(1) Physical data independence refers to the ability to modify the underlying database file system structures, indexes and access methods without having to modify the conceptual schema.

- (a) True
- (b) False

(2) The following things are commonly true about the state of the database:

- (a) It describes the design of the database schema at a specific point in time
- (b) It changes more frequently than the schema of the database
- (c) All of the above
- (d) None of the above

(3) The concept that most appropriately expresses the separation of the operations of the programs that we use from the specific ways that we structure and store the underlying data is known as program-metadata compilation.

- (a) True
- (b) False

(4) Which of the following statements are true?

- (a) SQL queries without DISTINCT always include duplicate entries in the result set
- (b) The presence of duplicates in the result set of an SQL query depends on the commands used within the query along with the use of keywords like ALL and DISTINCT
- (c) SQL is based on set theory and always removes duplicate entries from the result set
- (d) All of the above
- (e) None of the above

(5) Some of the main advantages of using relational database systems include:

- (a) SQL allows users to fundamentally ignore the underlying (and deeply technical) storage details of their data and (instead) focus on the conceptual structure of their data.
- (b) Relational database management systems often use special databases (e.g., catalogs) to store information (meta-data) about the structure of the other databases.
- (c) The three-schema architectures of many database systems are intended to allow you to change the low-level physical data structures and mapping details without having to change the higher-level conceptual schema or external views.
- (d) All of the above
- (e) None of the above

(6) Relational databases are often accessed using SQL, which is an imperative (i.e., procedural) language that allows the user to specifically direct how the database searches the records in the tables and assembles the result set.

- (a) True
- (b) False

(7) The information contained in the clauses of a Structured Query Language (SQL) query are normally "evaluated" (i.e., accessed) in the following sequence when the query is being processed (ignore ORDER BY for this problem):

- (a) SELECT – FROM – WHERE – GROUP BY – HAVING
- (b) FROM – GROUP BY – WHERE – HAVING – SELECT
- (c) FROM – WHERE – GROUP BY – HAVING – SELECT
- (d) SELECT – WHERE – GROUP BY – HAVING – FROM
- (e) FROM – WHERE – HAVING – GROUP BY – SELECT

(8) Databases are software packages/systems that can be used to create and manage one or more databases management systems.

- (a) True
- (b) False

Determine which of the following SQL queries are functionally correct for the given English requirements. A query is correct if (and only if) it would return the correct results for all valid database states – not just the current database state. Duplicate rows are OK unless explicitly forbidden by the query requirements.

(9) Display all of the airport information in the order of their country first (A-Z), and then with busier airports listed before "less busy" airports.

- (a) select * from airports order by country, traffic desc;
- (b) select * from airports order by country asc, traffic desc;
- (c) All of the above
- (d) None of the above

(10) Display all of the information for any flights (without duplicates) that depart from or arrive into the ATL airport.

- (a) select * from flights where depart_from = 'ATL' or arrive_at = 'ATL';
- (b) select * from flights where depart_from like '%ATL' or arrive_at like 'ATL%';
- (c) All of the above
- (d) None of the above

(11) Display all of the passenger information for those passengers carrying more than one bag.

- (a) select * from passengers where bags between 2 and 3;
- (b) select * from passengers where bags > 1;
- (c) All of the above
- (d) None of the above

(12) Display the total number of passengers on flight 52.

- (a) select count(*) from passengers where taking_flight like '%52%';
- (b) select sum(pid) from passengers where taking_flight in (52);
- (c) All of the above
- (d) None of the above

(13) Display the flight number and total number of bags for each flight that has at least one passenger.

- (a) select taking_flight, sum(bags) from passengers group by taking_flight;
- (b) select sum(bags) from passengers group by taking_flight;
- (c) All of the above
- (d) None of the above

(14) Display the list of flight identifiers without duplicates.

- (a) select distinct taking_flight from passengers;
- (b) (select taking_flight from passengers) union all (select taking_flight from passengers);
- (c) All of the above
- (d) None of the above

(15) Display all of the airport information for airports located in a country that begins with the word 'United'.

- (a) select * from airports where country like 'United_%';

(b) select * from airports where country like 'United%';

(c) All of the above

(d) None of the above

(16) Display the airline, flight number and duration of all flights departing from China.

(a) select airline, flight_num, duration from flights where depart_from in ('PEK', 'PVG');

(b) select airline, flight_num, duration from flights join airports on depart_from = apcode where country = 'China';

(c) All of the above

(d) None of the above

(17) Display the durations of the longest and shortest flights.

(a) select max(duration) as 'longest', min(duration) as 'shortest' from flights;

(b) (select 'longest' as 'category', max(duration) as 'duration' from flights) union (select 'shortest', min(duration) from flights);

(c) All of the above

(d) None of the above

(18) Display the total airport traffic for each country which has fewer than three airports.

(a) select country, count(traffic) from airports group by country having count(*) < 3;

(b) select country, sum(traffic) from airports where country in (select country from airports group by country having count(*) < 3) group by country;

(c) All of the above

(d) None of the above

(19) Display the airline and flight number for any flight that has at least one passenger.

(a) select airline, flight_num from flights, passengers group by airline, flight_num having count(*) >= 1;

(b) select airline, flight_num from flights, passengers where fid = taking_flight group by airline, flight_num having count(*) > 1;

(c) All of the above

(d) None of the above

(20) Display the total number of flights that cost more than \$700 for each airline, but only for airlines that have fewer than three flights.

(a) select airline, count(*) from flights where cost > 700 group by airline having count(*) < 3;

(b) select airline, count(*) from flights where cost > 700 and airline in (select airline from flights group by airline having count(*) < 3) group by airline;

(c) All of the above

(d) None of the above

(21) Display all passenger information for passengers on a flight that departed from the United States.

(a) select * from passengers where taking_flight in (select fid from flights where depart_from in (select apcode from airports where country = 'United States'));

(b) select * from passengers where taking_flight in (select fid from flights, airports where depart_from = apcode or country = 'United States');

(c) All of the above

(d) None of the above

(22) Display the airport code and the number of flights arriving at that airport.

All airports must be included in your result set, and '0' or 'null' values will be interpreted as zero flights.

(a) select apcode, count(*) from airports left outer join flights

- on apcode = arrive_at group by apcode, apname;
(b) select apcode, count(*) from flights left outer join airports
on apcode = arrive_at group by apcode, apname;
(c) All of the above
(d) None of the above

Determine the number of rows that will be produced by the following SQL queries.

(23) Display the flight identifier, airline, and flight number along with the number of passengers on that flight.

All flights must be included in your result set, and '0' or 'null' values will be interpreted as zero passengers.

- (a) select fid, airline, flight_num, total_riders from flights,
(select taking_flight, count(*) as 'total_riders' from passengers group by taking_flight) as temp
where fid = taking_flight;
(b) select fid, airline, flight_num, count(*) from flights left outer join passengers
on fid = taking_flight group by fid, airline, flight_num;
(c) All of the above
(d) None of the above

(24) select arrive_at, count(*) from flights group by arrive_at;

- (a) 0
(b) 6
(c) 7
(d) 8
(e) None of the above (also includes errors in query)

(25) select airline, depart_from, count(*) from flights where fid between 51 and 55 group by airline, depart_from;

- (a) 0
(b) 3
(c) 4
(d) 5
(e) None of the above (also includes errors in query)

(26) select taking_flight, count(*), sum(bags) from passengers where pid between 1 and 6
group by taking_flight having count(*) < 2;

- (a) 0
(b) 2
(c) 3
(d) 4
(e) None of the above (also includes errors in query)

(27) select distinct arrive_at from flights where airline not like '%China%' and airline not like '%Air%';

- (a) 0
(b) 4
(c) 5
(d) 6
(e) None of the above (also includes errors in query)

(28) (select airline from flights where depart_from = 'PVG') union (select airline from flights where arrive_at = 'PVG');

- (a) 0
- (b) 1
- (c) 2
- (d) 3
- (e) None of the above (also includes errors in query)

(29) select * from flights where depart_from in (select apcode from airports where country = 'China') and fid in (select taking_flight from passengers where bags > 1);

- (a) 0
- (b) 1
- (c) 2
- (d) 3
- (e) None of the above (also includes errors in query)

(30) select * from flights, passengers where bags > 0;

- (a) 0
- (b) 60
- (c) 70
- (d) 90
- (e) None of the above (also includes errors in query)

(31) select * from airports join flights on apcode = depart_from;

- (a) 0
- (b) 5
- (c) 10
- (d) 20
- (e) None of the above (also includes errors in query)

(32) select * from flights left outer join passengers on fid = taking_flight;

- (a) 0
- (b) 5
- (c) 10
- (d) 20
- (e) None of the above (also includes errors in query)

(33) select * from airports, flights where country in ('United States', 'United Arab Emirates') and (apcode = depart_from or apcode = arrive_at);

- (a) 0
- (b) 7
- (c) 9
- (d) 11
- (e) None of the above (also includes errors in query)

Any remaining space is yours for scratch work.

Use the following tables for all of the SQL related questions. These tables are part of a database about air travel. The air travel information below is all that you need to complete the questions.

This table is named **airports**, and each record contains facts about: a unique identifier for that airport (*apcode*); the name of the airport (*apname*), along with its location (i.e., *country*, *state*, and *city*); and, a relative measure of how busy the airport is in terms of passenger flow (*traffic*).

apcode	apname	country	state	city	traffic
ATL	Hartsfield Jackson	United States	Georgia	Atlanta	54.4
DXB	Dubai International	United Arab Emirates	Dubai	Garhoud	41.3
HND	Tokyo Haneda	Japan	Tokyo	Ota	41.4
LAX	Los Angeles	United States	California	Los Angeles	43.0
LHR	London Heathrow	United Kingdom	London	Hillingdon	38.8
ORD	Chicago OHare	United States	Illinois	Chicago	39.5
PEK	Beijing Capital	China	Beijing	Chaoyang-Shunyi	49.2
PVG	Shanghai Pudong	China	Shanghai	Pudong	38.2

This table is named **flights**, and each record contains facts about: a unique identifier for that flight (*fid*); the *airline* company and flight number (*flight_num*); the airports of departure (*depart_from*) and arrival (*arrive_at*); the *duration* of the flight including layovers; and, the *cost* of the flight.

fid	airline	flight_num	depart_from	arrive_at	duration	cost
50	Air China	7450	ATL	PEK	19	1051.00
51	United	7983	LAX	HND	13	913.00
52	Turkish Airlines	763_5	DXB	ORD	20	1296.00
53	Swiss Air	317_188	LHR	PVG	15	533.00
54	Delta	1735	ATL	LAX	5	604.00
55	United	660	LAX	ORD	4	451.00
56	China Eastern	271	PEK	PVG	2	468.00
57	United	1027	ORD	ATL	2	340.00
58	Emirates	303	PVG	DXB	11	877.00
59	Emirates	3	DXB	LHR	8	709.00

This table is named **passengers**, and each record contains facts about: a unique identifier for that passenger (*pid*); the first (*fname*) and last (*lname*) names of the passenger; the identifier of the flight that they are taking (*taking_flight*); and, the number of *bags* they are carrying.

pid	fname	lname	taking_flight	bags
1	Susanna	Delacruz	51	1
2	Bryony	Stark	52	3
3	Michael	Kayden	52	2
4	Kamila	Ferry	52	3
5	Abigayle	Velez	55	3
6	Laurence	McCormick	56	0
7	Willow	McDaniel	56	1
8	Portia	Banks	58	1
9	Samuel	Carpenter	59	0

Some key facts about the data, including how the tables are related:

- The *taking_flight* identifier values in the **passengers** table must represent a valid flight identifier (*fid*) from the **flights** table.

The airport identifier values (*depart_from* and *arrive_at*) in the **flights** table must represent valid airport identifiers (*apcode*) from the **airports** table.

Name (Last, First): _____

GT Username (e.g. gburdell3): _____

GT ID (e.g. 900-----): _____

Signature: _____

CS4400 – Introduction to Database Systems

Exam #1 – Database Basic Concepts, Three-Schema Architecture & Structured Query Language (SQL) DML

Versions [E, G, L, U]

- By writing your name at the top of this test **you are certifying that this test is entirely your own work per the Georgia Institute of Technology Academic Honor Code.**
- The exam is ***closed book*, *closed notes***, and ***no computer/calculator***.
- Using extra sheets of completely blank paper for scratch work is acceptable.
- Please **write the version of the Test Booklet on your Answer Sheet immediately**. You will lose points if you leave the version blank or write in the incorrect version letter.
- You must **turn in the Answer Sheet and the complete Test Booklet** to receive credit for the exam.
- Please **circle the final/official answer clearly in the Test Booklet** for each problem.
- You may ask the proctors for clarification but you are ultimately responsible for the answer you write on the paper.
- You may separate the pages if that makes it easier to work on the problems, but it's your responsibility to ensure that you reassemble and submit all Test Booklet pages before you leave.
- There are thirty-three (33) questions, but **you will only be graded on the basis of thirty (30) questions**. If you get more than 30 questions correct, you will keep the "bonus points."

Pace yourself, trust your studies, and good luck!

Versions [E, G, L, U]

(1) Logical data independence refers to the ability to modify the underlying conceptual schema without having to modify the external views.

- (a) True
- (b) False

(2) The following things are commonly true about the schema of the database:

- (a) It describes the data that is actually stored in the database at a specific point in time
- (b) It changes more frequently than the state of the database
- (c) All of the above
- (d) None of the above

(3) The concept that most appropriately expresses the separation of the operations of the programs that we use from the specific ways that we structure and store the underlying data is known as program-metadata compilation.

- (a) True
- (b) False

(4) Which of the following statements are true?

- (a) SQL is based on set theory and always removes duplicate entries from the result set
- (b) SQL queries without DISTINCT always include duplicate entries in the result set
- (c) The presence of duplicates in the result set of an SQL query depends on the commands used within the query along with the use of keywords like ALL and DISTINCT
- (d) All of the above
- (e) None of the above

(5) Some of the main advantages of using relational database systems include:

- (a) SQL requires users to understand the underlying (and deeply technical) storage details of their data in order to be able to query the data.
- (b) Relational database management systems require you to create structures like tables and views that cannot be changed once they have been created.
- (c) The three-schema architectures of many database systems are intended to allow you to change the low-level physical data structures and mapping details without having to change the higher-level conceptual schema or external views.
- (d) All of the above
- (e) None of the above

(6) Relational databases are often accessed using SQL, which is a declarative language that allows the user to specify the desired result set without having to direct how the database searches the tables.

- (a) True
- (b) False

(7) The information contained in the clauses of a Structured Query Language (SQL) query are normally "evaluated" (i.e., accessed) in the following sequence when the query is being processed (ignore ORDER BY for this problem):

- (a) SELECT – FROM – WHERE – GROUP BY – HAVING
- (b) FROM – WHERE – HAVING – GROUP BY – SELECT
- (c) SELECT – WHERE – GROUP BY – HAVING – FROM
- (d) FROM – GROUP BY – WHERE – HAVING – SELECT
- (e) FROM – WHERE – GROUP BY – HAVING – SELECT

(8) Database management systems are software packages/systems that can be used to create and manage one or more databases.

- (a) True
- (b) False

Determine which of the following SQL queries are functionally correct for the given English requirements. A query is correct if (and only if) it would return the correct results for all valid database states – not just the current database state. Duplicate rows are OK unless explicitly forbidden by the query requirements.

(9) Display all of the airport information in the order of their country first (A-Z), and then with busier airports listed before "less busy" airports.

- (a) select * from airports order by country, traffic desc;
- (b) select * from airports order by country asc, traffic desc;
- (c) All of the above
- (d) None of the above

(10) Display all of the information for any flights (without duplicates) that depart from or arrive into the ATL airport.

- (a) select * from flights where depart_from = 'ATL' or arrive_at = 'ATL';
- (b) (select * from flights where arrive_at = 'ATL') union (select * from flights where depart_from = 'ATL');
- (c) All of the above
- (d) None of the above

(11) Display all of the passenger information for those passengers carrying more than one bag.

- (a) select * from passengers where bags between 2 and 3;
- (b) select count(*) from passengers where bags > 1;
- (c) All of the above
- (d) None of the above

(12) Display the total number of passengers on flight 52.

- (a) select count(*) from passengers where taking_flight = 52;
- (b) select sum(pid) from passengers where taking_flight in (52);
- (c) All of the above
- (d) None of the above

(13) Display the flight number and total number of bags for each flight that has at least one passenger.

- (a) select distinct taking_flight from passengers;
- (b) select sum(bags) from passengers group by taking_flight;
- (c) All of the above
- (d) None of the above

(14) Display the list of flight identifiers without duplicates.

- (a) (select taking_flight from passengers) union all (select taking_flight from passengers);
- (b) select taking_flight from passengers;
- (c) All of the above
- (d) None of the above

(15) Display all of the airport information for airports located in a country that begins with the word 'United'.

- (a) select * from airports where country like 'United%';
- (b) select * from airports where country like '%United_%';
- (c) All of the above
- (d) None of the above

(16) Display the airline, flight number and duration of all flights departing from China.

- (a) select airline, flight_num, duration from flights where depart_from in ('PEK', 'PVG');
- (b) select airline, flight_num, duration from flights join airports on depart_from = apcode where country = 'China';
- (c) All of the above
- (d) None of the above

(17) Display the durations of the longest and shortest flights.

- (a) select max(duration) as 'longest', min(duration) as 'shortest' from flights;
- (b) (select 'longest' as 'category', max(duration) as 'duration' from flights) union (select 'shortest', min(duration) from flights);
- (c) All of the above
- (d) None of the above

(18) Display the total airport traffic for each country which has fewer than three airports.

- (a) select country, count(traffic) from airports group by country having count(*) < 3;
- (b) select sum(traffic) from airports group by country having count(*) < 3;
- (c) All of the above
- (d) None of the above

(19) Display the airline and flight number for any flight that has at least one passenger.

- (a) select airline, flight_num from flights where fid in (select taking_flight from passengers);
- (b) select airline, flight_num from flights, passengers group by airline, flight_num having count(*) >= 1;
- (c) All of the above
- (d) None of the above

(20) Display the total number of flights that cost more than \$700 for each airline, but only for airlines that have fewer than three flights.

- (a) select airline, count(*) from flights where cost > 700 group by airline having count(*) < 3;
- (b) select airline, count(*) from flights where cost > 700 and airline in (select airline from flights group by airline having count(*) < 3) group by airline;
- (c) All of the above
- (d) None of the above

(21) Display all passenger information for passengers on a flight that departed from the United States.

- (a) select * from passengers where taking_flight in (select fid from flights, airports where depart_from = apcode and country = 'United States');
- (b) select * from passengers where taking_flight in (select fid from flights where depart_from in (select apcode from airports where country = 'United States'));
- (c) All of the above
- (d) None of the above

(22) Display the airport code and the number of flights arriving at that airport.

All airports must be included in your result set, and '0' or 'null' values will be interpreted as zero flights.

- (a) select apcode, total_in from airports,

- (select arrive_at, count(*) as 'total_in' from flights group by arrive_at) as temp
where apcode = arrive_at;
(b) select apcode, count(*) from flights left outer join airports
on apcode = arrive_at group by apcode, apname;
(c) All of the above
(d) None of the above

Determine the number of rows that will be produced by the following SQL queries.

(23) Display the flight identifier, airline, and flight number along with the number of passengers on that flight.

All flights must be included in your result set, and '0' or 'null' values will be interpreted as zero passengers.

- (a) select fid, airline, flight_num, total_riders from flights,
(select taking_flight, count(*) as 'total_riders' from passengers group by taking_flight) as temp
where fid = taking_flight;
(b) (select fid, airline, flight_num, total_riders from flights,
(select taking_flight, count(*) as 'total_riders' from passengers group by taking_flight) as temp
where fid = taking_flight) union
(select fid, airline, flight_num, 0 from flights where fid not in (select taking_flight from passengers));
(c) All of the above
(d) None of the above

(24) select depart_from, count(*) from flights group by depart_from;

- (a) 0
(b) 6
(c) 7
(d) 8
(e) None of the above (also includes errors in query)

(25) select airline, depart_from, count(*) from flights where fid between 51 and 55 group by airline, depart_from;

- (a) 0
(b) 3
(c) 4
(d) 5
(e) None of the above (also includes errors in query)

(26) select taking_flight, count(*), sum(bags) from passengers where pid between 2 and 9
group by taking_flight having sum(bags) > 0;

- (a) 0
(b) 2
(c) 3
(d) 4
(e) None of the above (also includes errors in query)

(27) select distinct depart_from from flights where airline like '%China%' or airline like '%Air%';

- (a) 0
(b) 4
(c) 5
(d) 6
(e) None of the above (also includes errors in query)

(28) (select airline from flights where depart_from = 'ORD') union (select airline from flights where arrive_at = 'ORD');

- (a) 0
- (b) 1
- (c) 2
- (d) 3
- (e) None of the above (also includes errors in query)

(29) select * from flights where depart_from in (select apcode from airports where country = 'United States')

and fid in (select taking_flight from passengers where bags < 2);

- (a) 0
- (b) 1
- (c) 2
- (d) 3
- (e) None of the above (also includes errors in query)

(30) select * from airports, flights where country <> 'China';

- (a) 0
- (b) 60
- (c) 70
- (d) 80
- (e) None of the above (also includes errors in query)

(31) select * from flights join passengers on fid = taking_flight;

- (a) 0
- (b) 5
- (c) 10
- (d) 20
- (e) None of the above (also includes errors in query)

(32) select * from airports left outer join flights on apcode = depart_from;

- (a) 0
- (b) 5
- (c) 10
- (d) 20
- (e) None of the above (also includes errors in query)

(33) select * from airports, flights where country in ('United States', 'Japan')

and (apcode = depart_from or apcode = arrive_at);

- (a) 0
- (b) 7
- (c) 9
- (d) 11
- (e) None of the above (also includes errors in query)

Any remaining space is yours for scratch work.

Use the following tables for all of the SQL related questions. These tables are part of a database about air travel. The air travel information below is all that you need to complete the questions.

This table is named **airports**, and each record contains facts about: a unique identifier for that airport (*apcode*); the name of the airport (*apname*), along with its location (i.e., *country*, *state*, and *city*); and, a relative measure of how busy the airport is in terms of passenger flow (*traffic*).

apcode	apname	country	state	city	traffic
ATL	Hartsfield Jackson	United States	Georgia	Atlanta	54.4
DXB	Dubai International	United Arab Emirates	Dubai	Garhoud	41.3
HND	Tokyo Haneda	Japan	Tokyo	Ota	41.4
LAX	Los Angeles	United States	California	Los Angeles	43.0
LHR	London Heathrow	United Kingdom	London	Hillingdon	38.8
ORD	Chicago OHare	United States	Illinois	Chicago	39.5
PEK	Beijing Capital	China	Beijing	Chaoyang-Shunyi	49.2
PVG	Shanghai Pudong	China	Shanghai	Pudong	38.2

This table is named **flights**, and each record contains facts about: a unique identifier for that flight (*fid*); the *airline* company and flight number (*flight_num*); the airports of departure (*depart_from*) and arrival (*arrive_at*); the *duration* of the flight including layovers; and, the *cost* of the flight.

fid	airline	flight_num	depart_from	arrive_at	duration	cost
50	Air China	7450	ATL	PEK	19	1051.00
51	United	7983	LAX	HND	13	913.00
52	Turkish Airlines	763_5	DXB	ORD	20	1296.00
53	Swiss Air	317_188	LHR	PVG	15	533.00
54	Delta	1735	ATL	LAX	5	604.00
55	United	660	LAX	ORD	4	451.00
56	China Eastern	271	PEK	PVG	2	468.00
57	United	1027	ORD	ATL	2	340.00
58	Emirates	303	PVG	DXB	11	877.00
59	Emirates	3	DXB	LHR	8	709.00

This table is named **passengers**, and each record contains facts about: a unique identifier for that passenger (*pid*); the first (*fname*) and last (*lname*) names of the passenger; the identifier of the flight that they are taking (*taking_flight*); and, the number of *bags* they are carrying.

pid	fname	lname	taking_flight	bags
1	Susanna	Delacruz	51	1
2	Bryony	Stark	52	3
3	Michael	Kayden	52	2
4	Kamila	Ferry	52	3
5	Abigayle	Velez	55	3
6	Laurence	McCormick	56	0
7	Willow	McDaniel	56	1
8	Portia	Banks	58	1
9	Samuel	Carpenter	59	0

Some key facts about the data, including how the tables are related:

- The *taking_flight* identifier values in the **passengers** table must represent a valid flight identifier (*fid*) from the **flights** table.
- The airport identifier values (*depart_from* and *arrive_at*) in the **flights** table must represent valid airport identifiers (*apcode*) from the **airports** table.