Faculty of Science

Name:
Student ID:
Course: CSCI 3030U Database Systems and Concepts
Submission: via Canvas, **pdf only** – paste diagrams and answers into this document

Instructions

- Use a legible color for your answers' text.
- You can access any externals resources or course materials.
- The instructor will answer questions via Slack (DM), do **NOT** post your question on public channels.
- This is an **individual** assessment. Do **NOT** copy from others, and do **NOT** share your answer with others. Remember, copying and pasting from online resources is also considered plagiarism. Any form of plagiarism will be treated accordingly to the university's code of conduct.

Before your start: get to know your data

Below you can see the collection of 3 tables for the **CustomOrder** database.

Table: **Agents**

AGENT_CODE	AGENT_NAME	WORKING_AREA	COMMISSION	PHONE_NO	COUNTRY
A007	Ramasundar	Bangalore	0.10	077-25815544	
A003	Alex	London	0.13	075-12458969	
A008	Alford	New York	0.12	044-25874365	
A011	Ravi Kumar	Bangalore	0.18	077-45625874	
A010	Santakumar	Chennai	0.14	007-22388644	
A012	Lucida	San Jose	0.12	044-52981425	
A005	Anderson	Brisbane	0.11	045-21447739	
A001	Subbarao	Bangalore	0.14	077-12346674	
A002	Mukesh	Mumbai	0.16	029-12365897	
A006	McDen	London	0.15	078-22255588	
A004	Ivan	Toronto	0.15	008-22544166	
A009	Benjamin	Hampshire	0.13	008-22536178	

Table: **Customers**

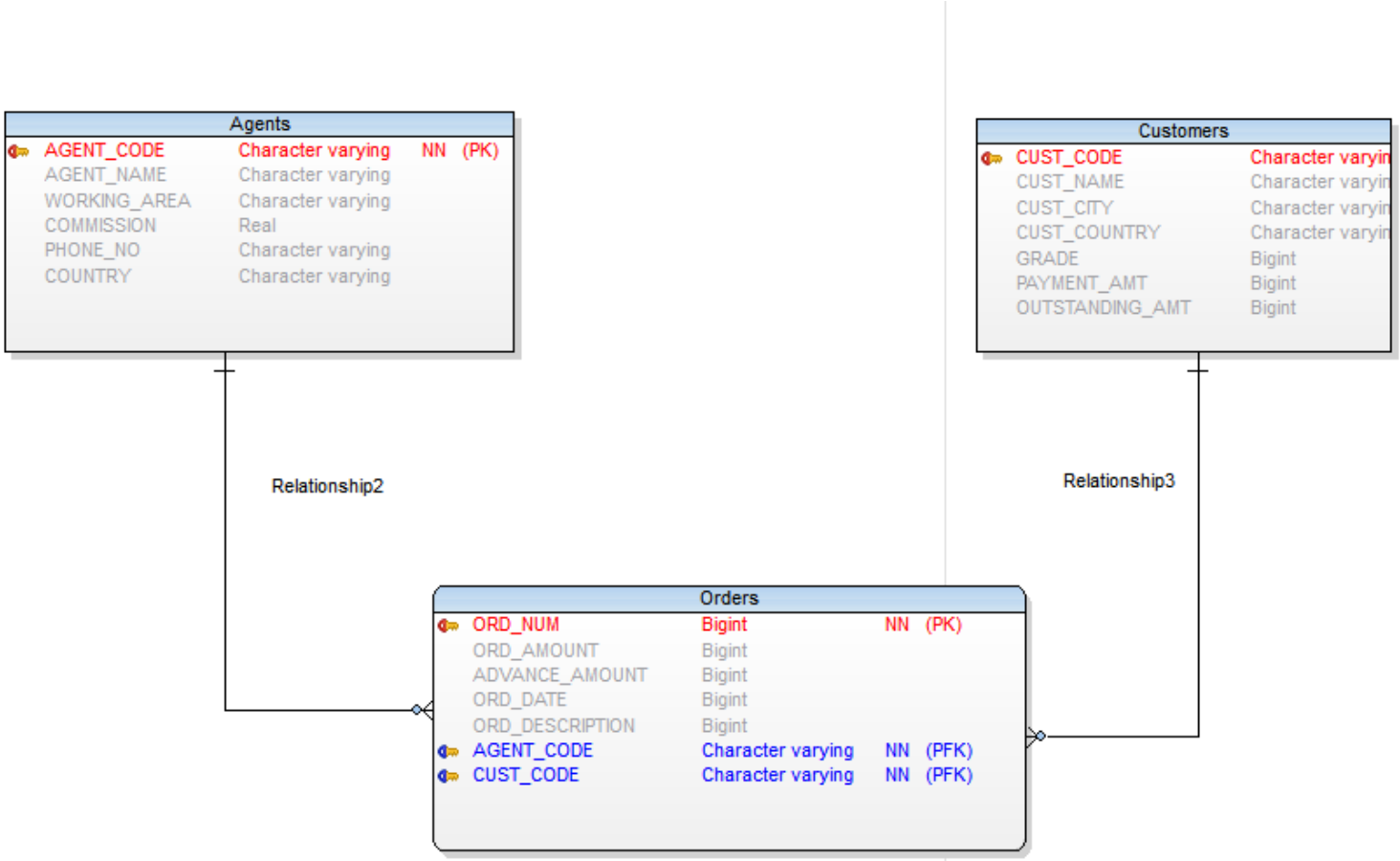
CUST_CODE	CUST_NAME	CUST_CITY	CUST_COUNTRY	GRADE	PAYMENT_AMT	OUTSTANDING_AMT
C00013	Holmes	London	UK	2	7000	4000
C00001	Michael	New York	USA	2	2000	6000
C00020	Albert	New York	USA	3	6000	6000
C00025	Ravindran	Bangalore	India	2	4000	8000
C00024	Cook	London	UK	2	7000	6000
C00015	Stuart	London	UK	1	3000	11000
C00002	Bolt	New York	USA	3	9000	3000
C00018	Fleming	Brisbane	Australia	2	9000	5000
C00021	Jacks	Brisbane	Australia	1	7000	7000
C00019	Yearannaidu	Chennai	India	1	7000	8000
C00005	Sasikant	Mumbai	India	1	7000	11000
C00007	Ramanathan	Chennai	India	1	9000	9000
C00022	Avinash	Mumbai	India	2	9000	9000
C00004	Winston	Brisbane	Australia	1	7000	6000
C00023	Karl	London	UK	0	7000	3000
C00006	Shilton	Toronto	Canada	1	6000	11000
C00010	Charles	Hampshire	UK	3	5000	5000
C00017	Srinivas	Bangalore	India	2	3000	9000
C00012	Steven	San Jose	USA	1	9000	3000
C00008	Karolina	Toronto	Canada	1	9000	5000
C00003	Martin	Toronto	Canada	2	7000	8000
C00009	Ramesh	Mumbai	India	3	3000	12000
C00014	Rangarappa	Bangalore	India	2	7000	12000
C00016	Venkatpati	Bangalore	India	2	7000	12000
C00011	Sundariya	Chennai	India	3	7000	11000

Table: **Orders**

ORD_NUM	ORD_AMOUNT	ADVANCE_AMOUNT	ORD_DATE	CUST_CODE	AGENT_CODE	ORD_DESCRIPTION
200100	1000	600	8/1/2008	C00013	A003	SOD
200110	3000	500	4/15/2008	C00019	A010	SOD
200107	4500	900	8/30/2008	C00007	A010	SOD
200112	2000	400	5/30/2008	C00016	A007	SOD
200113	4000	600	6/10/2008	C00022	A002	SOD
200102	2000	300	5/25/2008	C00012	A012	SOD
200114	3500	2000	8/15/2008	C00002	A008	SOD
200122	2500	400	9/16/2008	C00003	A004	SOD
200118	500	100	7/20/2008	C00023	A006	SOD
200119	4000	700	9/16/2008	C00007	A010	SOD
200121	1500	600	9/23/2008	C00008	A004	SOD
200130	2500	400	7/30/2008	C00025	A011	SOD
200134	4200	1800	9/25/2008	C00004	A005	SOD
200108	4000	600	2/15/2008	C00008	A004	SOD
200103	1500	700	5/15/2008	C00021	A005	SOD
200105	2500	500	7/18/2008	C00025	A011	SOD
200109	3500	800	7/30/2008	C00011	A010	SOD
200101	3000	1000	7/15/2008	C00001	A008	SOD
200111	1000	300	7/10/2008	C00020	A008	SOD
200104	1500	500	3/13/2008	C00006	A004	SOD
200106	2500	700	4/20/2008	C00005	A002	SOD
200125	2000	600	10/10/2008	C00018	A005	SOD
200117	800	200	10/20/2008	C00014	A001	SOD
200123	500	100	9/16/2008	C00022	A002	SOD
200120	500	100	7/20/2008	C00009	A002	SOD
200116	500	100	7/13/2008	C00010	A009	SOD
200124	500	100	6/20/2008	C00017	A007	SOD
200126	500	100	6/24/2008	C00022	A002	SOD
200129	2500	500	7/20/2008	C00024	A006	SOD
200127	2500	400	7/20/2008	C00015	A003	SOD
200128	3500	1500	7/20/2008	C00009	A002	SOD
200135	2000	800	9/16/2008	C00007	A010	SOD
200131	900	150	8/26/2008	C00012	A012	SOD
200133	1200	400	6/29/2008	C00009	A002	SOD

Part 1 – Represent your database (2 points)

Attach here an Entity Relationship Diagram of the given **CustomOrder** database.
You may use any diagramming software or hand-drawn image (note that it should be legible). Your diagram should include all the keys and relationships between the relations including multiplicity. You don't need to include all attributes in each table, only the ones you judge the most relevant.



Part 2 – Creating your database (1 point)

Write the SQL statement to create the relation **Orders**. Including all attributes.

```
CREATE TABLE "Orders"
(
  "ORD_NUM" Bigint NOT NULL,
  "ORD_AMOUNT" Bigint,
  "ADVANCE_AMOUNT" Bigint,
  "ORD_DATE" Date,
  "ORD_DESCRIPTION" Bigint,
  "AGENT_CODE" Character varying NOT NULL,
  "CUST_CODE" Character varying NOT NULL
)
WITH (
  autovacuum_enabled=true)
;

ALTER TABLE "Orders" ADD CONSTRAINT "PK_Orders" PRIMARY KEY ("ORD_NUM","AGENT_CODE","CUST_CODE")
;

-- Create foreign keys (relationships) section ------------------------------------------------

ALTER TABLE "Orders"
  ADD CONSTRAINT "Relationship2"
    FOREIGN KEY ("AGENT_CODE")
    REFERENCES "Agents" ("AGENT_CODE")
      ON DELETE NO ACTION
      ON UPDATE NO ACTION
;
```

```
ALTER TABLE "Orders"
  ADD CONSTRAINT "Relationship3"
    FOREIGN KEY ("CUST_CODE")
    REFERENCES "Customers" ("CUST_CODE")
      ON DELETE NO ACTION
      ON UPDATE NO ACTION
;
```

Part 3 – Understanding the data (3 points – 0.6 each)

In this part, you will be using SQL queries to answer questions about your data.
For each of the items below, please write the SQL query(ies) AND paste the expected output from your query:
Ps.: for calculated values you may list them as the formula rather than calculating the actual value. E.g., "column_value"*2.

a) How much commission has each agent made based on the orders' amount? Result (name, commission).

```
select Agent.AGENT_NAME, (Agent.COMMISSION * Order.ORD_AMOUNT) as commission from AGENTS join ORDERS on
Agents.AGENT_CODE = Order.AGENT_CODE
```

b) List the clients whose outstanding balance is double or more than their payment amount as "RED", otherwise as "BLUE" in their status. Result (client_code, name, status)

```
select  CUST_CODE as client_code, CUST_NAME as name,
CASE
when (PAYMENT_AMT * 2 <= OUTSTANDING_AMT)
then 'RED'
else 'BLUE'
end
from Customers
```

c) Considering the previous data query results, list all the agents' contact information that work in the area of the clients on "RED" status. Result (agent_code, name, phone)

```
Select Agent.AGENT_CODE as agent_code, Agent.AGENT_NAME as name, Agent.PHONE_NO as phone join Orders on
Agent.AGENT_CODE = Order.AGENT_CODE JOIN (select  CUST_CODE as client_code, CUST_NAME as name,
case
when (PAYMENT_AMT * 2 <= OUTSTANDING_AMT)
then 'RED'
else 'BLUE'
```

```
end as status
from Customers) y
where y.status = 'RED'
```

d) List the orders from April sorted by agent and amount. Result (date, order_num, agent_name, client_name, order_amount, pending_amount)

```
Select Orders.ORD_DATE as date, ORD_NUM as  order_num, Agent.AGENT_NAME as agent_name, Customers.CUST_NAME as client_name, Orders.ORD_AMOUNT as order_amount
From Agents
Join Orders on
Agents.AGENT_CODE = Orders.AGENT_CODE
where EXTRACT(MONTH FROM Orders.DATE) = 4
order by "AGENT_CODE", "ORD_AMOUNT"desc
```

e) Which country has the greatest number of customers? What is the average of both outstanding and payment amount for this country? Result (country, number_clients, avg_outstanding, avg_payment)

```
Select "Country", Count("Country")
From "Customers"
Group by "Country" desc
```

Part 4 – Advanced database operations and modifications (3 points – 1 point each)

In this part, you will be writing SQL query(ies) to satisfy each of the items below:

a) Based on the Customers table values, fill the column Agents.Country with its respective region's country or NULL if not found.

```
update "Agents"
set Agents.COUNTRY = Customers.COUNTRY
where Agents.WORKING_AREA = Customers.CUST_CITY
```

b) Create a new column **preferred_agent** in the Customers table and store the agent_code of agent who handled their most recent order or NULL if no orders are found for this customer.

```
alter table "Customers"
add "preferred_agent" Character varying
```

c) Create a virtual view "CUST_INDIA" of all customers in India (CUST_CODE, CUST_NAME, CUST_REGION, GRADE)

```
create view "CUST_INDIA" AS
select "CUST_CODE", "CUST_NAME", "CUST_REGION", "GRADE"
from "Customers"
where "Country" = 'India'
```

Part 5 – Integrating SQL and implementation (1 point, 0.25 each gap)

Consider the following Java code snippet and fill the gaps:

```
…
//statement that selects orders from a month and including the agent's commission rate for each order
statement = con.prepareStatement(select Agents.COMMISSION, Order.ORD_AMOUNT from Orders
join Agents on Orders.AGENT_CODE = Agents.AGENT_CODE where Orders.Date);
statement.setInt(1, month);

ResultSet rs = statement.executeQuery();

while (rs.hasNext()) {
        agent_rate = rs.getDouble(1);
        order_amount = rs. getInt(2);
        System.out.println(agent_rate*order_amount);
    }
…
```