HABIB UNIVERSITY

Database Systems

CS/CE 355/373 Fall 2023

Instructor: Maria Samad

Design a Relational Database Solution

Assume you have the following database systems:

- Bank Management System
- Airline Reservation System
- Inventory Management System

Choose any one of the above examples (or if you want, you can pick any other of your own choice as well), and design a relational database system for it with the following minimum requirements

1. Your DBS should at least have 3 tables/relations relevant to the chosen idea
2. Define Metadata for the relations (from part 1), by making sure there are at least 2 attributes in each table
3. Specify the ***key*** in each of the tables – remember, keys are fields with unique values that are used for accessing a particular record in each table
4. Create and populate the tables with at least 3 records by following the constraints, as specified in part 2
5. Define at least **ONE** master transaction for the chosen database, by specifying which data item is being accessed from which table – this is like writing a query, but as we don't know the syntax yet, so use simple statements to explain

ONE POSSIBLE SOLUTION:

- ***Bank Management System:***
  1. Relations/Tables:

ACCOUNT:

ACCOUNT_ID	ACCOUNT_TYPE	DATE_OPENED	BALANCE	CUSTOMER_ID

CUSTOMER:

CUST_ID	CUST_NAME	CUST_PHONE	CUST_EMAIL	OPENING_DATE

TRANSACTIONS:

TRANS_ID	TRANS_DATE	TRANS_AMOUNT	ACCOUNT_ID	PURCHASE_ID

  2. Metadata:

RELATIONS:

RELATION_NAME	NUM_COLUMNS
ACCOUNT	5
CUSTOMER	5
TRANSACTIONS	5

COLUMNS:

COLUMN_NAME	DATA_TYPE	BELONGS_TO_RELATION
Account_Id	Integer (10)	ACCOUNT
Account_Type	Character (30)	ACCOUNT
Date_Opened	Date	ACCOUNT
Balance	Float (20,2)	ACCOUNT
Customer_Id	Integer (10)	ACCOUNT
Cust_Id	Integer (10)	CUSTOMER
Cust_Name	Character (30)	CUSTOMER
Cust_Phone	Integer (10)	CUSTOMER
Cust_Email	Character (40)	CUSTOMER
Opening_Date	Date	CUSTOMER
Trans_Id	Integer (10)	TRANSACTION
Trans_Date	Date	TRANSACTION
Trans_Amount	Float (8,2)	TRANSACTION
Account_Id	Integer (10)	TRANSACTION
Purchase_Id	Integer (10)	TRANSACTION

3. Keys are identified in tables in part 1 – The underlined column names are taken as primary key in this situation

4. Some values in the tables

ACCOUNT:

ACCOUNT_ID	ACCOUNT_TYPE	DATE_OPENED	BALANCE	CUSTOMER_ID
1230984567	Chequing	20-09-1998	357000579.25	4320091980
5186293407	Savings	15-07-2005	75120345549.99	1234567890
6070809000	Savings	14-06-2007	999999999999.00	2614062007
3390871233	Minor	14-06-2012	369123.55	2614062007
1020304050	Chequing	12-01-2013	250000.00	1234567890

CUSTOMER:

CUST_ID	CUST_NAME	CUST_PHONE	CUST_EMAIL	OPENING_DATE
4320091980	Michael Jordan	2265054007	michael_jordan@nba.com	20-09-1998
1234567890	Roger Federer	3002225555	roger_federer@tennis.com	15-07-2005
2614062007	Christiano Ronaldo	6138302547	cr7@fifa.com	14-06-2007

TRANSACTIONS:

TRANS_ID	TRANS_DATE	TRANS_AMOUNT	ACCOUNT_ID	PURCHASE_ID
1286420191	13-03-2022	620.25	1230984567	1212121212
1286420192	13-03-2022	10.15	1020304050	2323232323
1286420193	15-04-2022	100.00	5186293407	3434343434

5. Christiano Ronaldo purchases a villa worth 5 million dollars in Riyadh on January 20th, 2023, which he paid for with his Savings account.
    - First access CUSTOMER Table to get Customer_ID of Christiano Ronaldo
        - Output = 2614062007
    - Then using the Customer_ID from above step, search in ACCOUNT Table to get the Account details for this customer
        - The search returns two results based on Customer_ID:

ACCOUNT_ID	ACCOUNT_TYPE	DATE_OPENED	BALANCE
6070809000	Savings	14-06-2007	999999999999.00
3390871233	Minor	14-06-2012	369123.55

        - From these results, get Account_Id and Balance for Account_Type = Savings
            - Account_Id = 6070809000, Balance = 999999999999.00
    - If the balance obtained above is greater than 5 million, i.e. 5000000, then enter a new transaction in the TRANSACTION Table for the purchasing. Assuming Purchase_ID = 4545454545. The Table will be updated as follows:

TRANSACTIONS:

TRANS_ID	TRANS_DATE	TRANS_AMOUNT	ACCOUNT_ID	PURCHASE_ID
1286420191	13-03-2022	620.25	1230984567	1212121212
1286420192	13-03-2022	10.15	1020304050	2323232323
1286420193	15-04-2022	100.00	5186293407	3434343434
1286420194	20-01-2023	5000000.00	6070809000	4545454545

    - However, the ACCOUNT Table will also need to be updated, as the Balance field will need to be adjusted. The updated ACCOUNT Table is:

ACCOUNT:

ACCOUNT_ID	ACCOUNT_TYPE	DATE_OPENED	BALANCE	CUSTOMER_ID
1230984567	Chequing	20-09-1998	357000579.25	4320091980
5186293407	Savings	15-07-2005	75120345549.99	1234567890
6070809000	Savings	14-06-2007	999994999999.00	2614062007
3390871233	Minor	14-06-2012	369123.55	2614062007
1020304050	Chequing	12-01-2013	250000.00	1234567890

- ***Airline Reservation System:***
  1. Relations/Tables:

FLIGHT:

Flight_ID	Origin	Destination	Departure_Date	Departure_Time	Arrival_Date	Arrival_Time

PASSENGER:

Pass_ID	Pass_Name	Passport_ID	Visa_Type	Flight_ID	Payment_ID

PAYMENT:

Payment_ID	Payment_Date	Amount	Method_Payment	Passenger_ID

  2. Metadata:

RELATIONS:

RELATION_NAME	NUM_COLUMNS
FLIGHT	7
PASSENGER	6
PAYMENT	5

COLUMNS:

COLUMN_NAME	DATA_TYPE	BELONGS_TO_RELATION
Flight_ID	Character (6)	FLIGHT
Origin	Character (30)	FLIGHT
Destination	Character (30)	FLIGHT
Departure_Date	Date	FLIGHT
Departure_Time	Time	FLIGHT
Arrival_Date	Date	FLIGHT
Arrival_Time	Time	FLIGHT
Pass_ID	Integer (10)	PASSENGER
Pass_Name	Character (40)	PASSENGER
Passport_ID	Character (6)	PASSENGER
Visa_Type	Character (10)	PASSENGER
Flight_ID	Character (6)	PASSENGER
Payment_ID	Integer (10)	PASSENGER
Payment_ID	Integer (10)	PAYMENT
Payment_Date	Date	PAYMENT
Amount	Float (10,2)	PAYMENT
Method_Payment	Character (10)	PAYMENT
Passenger_ID	Integer (10)	PAYMENT

  3. Keys are identified in tables in part 1 – The underlined column names are taken as primary key in this situation

4. Some values in the tables

FLIGHT:

Flight_ID	Origin	Destination	Departure_Date	Departure_Time	Arrival_Date	Arrival_Time
EK1234	Karachi	Toronto	08-09-2023	21:10	09-09-2023	09:45
PK3456	Frankfurt	Lahore	10-09-2023	05:55	10-09-2023	23:15
EY4567	Istanbul	Islamabad	15-09-2023	12:35	15-09-2023	21:45
TG6789	Sydney	Karachi	20-09-2023	18:30	21-09-2023	10:00

PASSENGER:

Pass_ID	Pass_Name	Passport_ID	Visa_Type	Flight_ID	Payment_ID
4321	Aubrey Drake	AD8901	Citizen	EK1234	35
6712	JR Oppenheimer	JR7123	Visit	PK3456	39
9823	Ben Stokes	BS7623	Work Permit	TG6789	45

PAYMENT:

Payment_ID	Payment_Date	Amount	Method_Payment	Passenger_ID
39	25-06-2023	1173.00	Cash	6712
35	30-08-2023	2491.00	Credit Card	4321
45	15-07-2023	1247.00	Cash	9823

5. Nusret Gökçe has decided to open a branch of Nusr-Et in Islamabad, for which he needs to travel from Istanbul to Islamabad for a survey. The government of Pakistan has issued him a Diplomat visa in honor of Pakistan-Turkey friendship. He will have to be in Islamabad before September 20th, 2023, and his manager will be using the company's credit card to pay for his flight.
    - First access FLIGHT Table to see if there is a flight available from Istanbul to Islamabad, and get its record so that its booking can be made

Flight_ID	Origin	Destination	Departure_Date	Departure_Time	Arrival_Date	Arrival_Time
EY4567	Istanbul	Islamabad	15-09-2023	12:35	15-09-2023	21:45

    - From this record, check if the flight reaches Islamabad before September 20th, 2023, and get its Flight_ID
        - Output: Flight_ID = EY4567
    - Then to book the flight, we will access PASSENGER Table to register Nusret Gökçe first. The table will be updated for him as follows:

PASSENGER:

Pass_ID	Pass_Name	Passport_ID	Visa_Type	Flight_ID	Payment_ID
4321	Aubrey Drake	AD8901	Citizen	EK1234	35
6712	JR Oppenheimer	JR7123	Visit	PK3456	39
9823	Ben Stokes	BS7623	Work Permit	TG6789	45
2456	Nusret Gökçe	NG3090	Diplomat	EY4567	56

    - Also, the payment needs to be confirmed, so update the PAYMENT Table with this new booking information

PAYMENT:

Payment_ID	Payment_Date	Amount	Method_Payment	Passenger_ID
39	25-06-2023	1173.00	Cash	6712
35	30-08-2023	2491.00	Credit Card	4321
45	15-07-2023	1247.00	Cash	9823
56	30-08-2023	772.00	Credit Card	2456

- ***Inventory Management System:***

1. Relations/Tables:

PRODUCT:

Prod_ID	Barcode	Prod_Name	Prod_Category	Prod_Price

PROVIDER:

Prov_ID	Prov_Name	Prov_Address	Prov_Website	Product_ID

INVENTORY:

Inventory_ID	Product_ID	Quantity_Available	Location

ORDER:

Order_ID	Order_Date	Provider_ID	Quantity_Ordered	Inventory_ID

2. Metadata:

RELATIONS:

RELATION_NAME	NUM_COLUMNS
PRODUCT	5
PROVIDER	5
INVENTORY	4
ORDER	6

COLUMNS:

COLUMN_NAME	DATA_TYPE	BELONGS_TO_RELATION
Prod_ID	Character (6)	PRODUCT
Barcode	Integer (12)	PRODUCT
Prod_Name	Character (50)	PRODUCT
Prod_Category	Character (30)	PRODUCT
Prod_Price	Float (8, 2)	PRODUCT
Prov_ID	Integer (3)	PROVIDER
Prov_Name	Character (30)	PROVIDER
Prov_Address	Character (40)	PROVIDER
Prov_Website	Character (30)	PROVIDER
Product_ID	Character (6)	PROVIDER
Inventory_ID	Integer (5)	INVENTORY
Product_ID	Character (6)	INVENTORY
Quantity_Available	Integer (4)	INVENTORY
Location	Character (10)	INVENTORY
Order_ID	Integer (10)	ORDER
Order_Date	Date	ORDER
Provider_ID	Integer (3)	ORDER
Quantity_Ordered	Integer (10)	ORDER
Inventory_ID	Integer (5)	ORDER

3. Keys are identified in tables in part 1 – The underlined column names are taken as primary key in this situation

4. Some values in the tables

PRODUCT:

Prod_ID	Barcode	Prod_Name	Prod_Category	Prod_Price
BK0001	908070605040	Database System Concepts	Book	99.99
HB9427	312019089786	Trezor	Handbag	49.98
BK0002	872917316495	Think Python	Book	55.50

PROVIDER:

Prov_ID	Prov_Name	Prov_Address	Prov_Website	Product_ID
135	McGraw-Hill	1325 6th Ave, New York, USA	https://www.mheducation.com/	BK0001
212	O'Reilly	PO Box 722, Farnham, UK	https://www.oreilly.com/	BK0002
348	Aldo	905 Rue Hodge, Saint-Laurent, Canada	https://www.aldoshoes.com/ca/en	HB9427

INVENTORY:

Inventory_ID	Product_ID	Quantity_Available	Location
732	HB9427	25	First floor
390	BK0001	50	Ground floor
504	BK0002	5	Ground floor

ORDER:

Order_ID	Order_Date	Provider_ID	Quantity_Ordered	Inventory_ID
287	01-01-2023	135	30	390
398	01-01-2023	212	40	504
409	14-06-2023	348	25	732

5. The new academic year has started, and there are around 100 students enrolled in the first year. They will all be provided with **Think Python** textbook from O'Reilly Learning. However, the inventory only contains 5 of them, so we need more, before October 1st, 2023. The shipment takes around 45 days, so the books must be ordered right away. Otherwise, the inventory will become useless after the deadline has passed. Calculate the total cost at the end.
    - First access PRODUCT Table to see if the desired product is already a part of inventory or not. If it wasn't, the PRODUCT Table will need to be updated first by assigning new Product ID to it. However, in this case, it is already present, so we extract the Product_ID and its price
        - Output: Prod_ID = BK0002, Prod_Price = $55.50
    - Using Product ID, check the Inventory if there are 100 copies present, i.e. access the entries with Product ID = BK0002, and get the Inventory ID and Quantity
        - Output: Inventory_ID = 504, Quantity = 5
    - The quantity is not sufficient, and we need to order 95 more copies. Using the Product ID, get the provider's information, so the order can be placed.
        - Output: Prov_ID = 212
    - Now place an order, which means adding a new record in the ORDER Table. As the Order_ID is generated at runtime, so assume it is 510, and Order_Date is current date. The table will be updated as follows:

ORDER:

Order_ID	Order_Date	Provider_ID	Quantity_Ordered	Inventory_ID
287	01-01-2023	135	30	390
398	01-01-2023	212	40	504
409	14-06-2023	348	25	732
510	30-08-2023	212	95	504

    - Total Cost can be calculated by multiplying Prod_Price (as extracted in the beginning) with the quantity, i.e. Cost = 95 * 55.50 = $5272.50