

Database Systems Coursework 2

PG Group 13
Matvey Ivashko
Tazim Chowdhury
Kyle Benoit
Athina Skourta

1. Relational Schema from coursework 1:

Main Guest

Primary key: MainGuestID

<u>MainGuestID</u>	Name	NumberGuests
41111	John Smith	0
42222	Mark Seth	1
43333	Ross Geller	2

AddGuest 1

Primary Key: AddID

<u>AddID</u>	Name
42111	Anthony Phillips
43111	Rachel Green
43313	Sarah White

AddGuest 2

Primary Key: AddID

Foreign Key: MainGuestID

<u>AddID</u>	MainGuestID
42111	42222
43111	43333
43313	43333

Rooms

Primary Key: RoomID

Foreign Key: MainGuestID

<u>RoomID</u>	MainGuestID	Type
111	41111	Excellent
122	42222	Deluxe
133	43333	Magnificent

Room Type

Primary Key: Type

Foreign Key: MainGuestID

<u>Type</u>	BasicPrice	MainGuestID
Excellent	1000	41111
Deluxe	2000	42222
Magnificent	3000	43333

Booking

Primary Key: BookingReference

Foreign Key : MainGuestID

<u>BookingReference</u>	StartDate	MainGuestID
11141111	15 Feb 2017	41111
12242222	21 June 2018	42222
13343333	7 December 2018	43333

FinalPrice

Primary Key: BookingReference

<u>BookingReference</u>	Price	NoNights
11141111	7000	7
12242222	11000	5
13343333	43200	12

RoomBooking

Primary Key: BookingReference

Foreign Keys: RoomID, MainGuestID

<u>BookingReference</u>	RoomID	MainGuestID
11141111	111	41111
12242222	122	42222
13343333	133	43333

2. The list of “create table” we used:

```
CREATE TABLE MainGuest
(MainGuestID NUMBER(5) NOT NULL PRIMARY KEY,
Name VARCHAR2(20),
NumberGuests NUMBER(2),
);
```

```
CREATE TABLE AddGuest1
(AddID NUMBER(5) NOT NULL PRIMARY KEY,
AddName VARCHAR2(20)
);
```

```
CREATE TABLE AddGuest2
(AddID NUMBER(5) NOT NULL PRIMARY KEY,
MainGuestID NUMBER(5),
FOREIGN KEY(MainGuestID) REFERENCES MainGuest(MainGuestID)
);
```

```
CREATE TABLE Rooms
(RoomID NUMBER(3) NOT NULL PRIMARY KEY,
Types VARCHAR2(12),
MainGuestID NUMBER(5),
FOREIGN KEY(MainGuestID) REFERENCES MainGuest(MainGuestID)
);
```

```
CREATE TABLE RoomsType
(Types VARCHAR2(13),
BasicPrice NUMBER(5)
);
```

```
CREATE TABLE Booking
(BookingReference NUMBER(10) NOT NULL PRIMARY KEY,
StartDate DATE,
MainGuestID NUMBER(5),
FOREIGN KEY(MainGuestID) REFERENCES MainGuest(MainGuestID)
);
```

```
CREATE TABLE FinalPrice
(BookingReference NUMBER(10) NOT NULL PRIMARY KEY,
Price NUMBER(7),
NoNights NUMBER(3)
);
```

```
CREATE TABLE RoomBooking
(BookingReference NUMBER(10) NOT NULL PRIMARY KEY,
RoomID NUMBER(3),
MainGuestID NUMBER(5),
FOREIGN KEY(MainGuestID) REFERENCES MainGuest(MainGuestID),
FOREIGN KEY(RoomID) REFERENCES Rooms(RoomID)
);
```

3. The sample test data:

```
SET TERMOUT ON
PROMPT Building demonstration tables. Please wait.
SET TERMOUT OFF
```

```
DROP TABLE MainGuest;
DROP TABLE AddGuest1;
DROP TABLE AddGuest2;
DROP TABLE Rooms;
DROP TABLE RoomsType;
DROP TABLE Booking;
DROP TABLE FinalPrice;
DROP TABLE RoomBooking;
```

```
CREATE TABLE MainGuest
(MainGuestID NUMBER(5) NOT NULL PRIMARY KEY,
Name VARCHAR2(20),
NumberGuests NUMBER(2),
);
```

```
INSERT INTO MainGuest VALUES
(41111,'John Smith',0);
INSERT INTO MainGuest VALUES
(42222,'Mark Seth',1);
INSERT INTO MainGuest VALUES
(43333,'Ross Geller',2);
INSERT INTO MainGuest VALUES
(44444,'Sara Chamberlain',2);
INSERT INTO MainGuest VALUES
(45555,'Theodore Russel',2);
INSERT INTO MainGuest VALUES
(46666,'Kyle Benoit',1);
INSERT INTO MainGuest VALUES
(47777,'Matthew Johnson',0);
INSERT INTO MainGuest VALUES
(48888,'Athena Scourta',2);
INSERT INTO MainGuest VALUES
(49999,'Tazim Chowdhury',0);
INSERT INTO MainGuest VALUES
(51111,'Cynthia Garrick',1);
INSERT INTO MainGuest VALUES
(52222,'Sophie Smith',3);
INSERT INTO MainGuest VALUES
```

```
(53333,'Pauline Duncan',3);
INSERT INTO MainGuest VALUES
(54444,'Tony Trump',2);
```

```
CREATE TABLE AddGuest1
(AddID NUMBER(5) NOT NULL PRIMARY KEY,
AddName VARCHAR2(20)
);
```

```
INSERT INTO AddGuest1 VALUES
(42111, 'Anthony Phillips');
INSERT INTO AddGuest1 VALUES
(43111,'Rachel Green');
INSERT INTO AddGuest1 VALUES
(43313,'Sarah White');
INSERT INTO AddGuest1 VALUES
(44111,'Atif Ali');
INSERT INTO AddGuest1 VALUES
(44222,'Shaheen Dhokia');
INSERT INTO AddGuest1 VALUES
(45111,'Chris Feynman');
INSERT INTO AddGuest1 VALUES
(45222,'Ekram Rizvi');
INSERT INTO AddGuest1 VALUES
(46111,'Sajida Rose');
INSERT INTO AddGuest1 VALUES
(48111,'Demitri Iacovou');
INSERT INTO AddGuest1 VALUES
(48222,'Pauline Kyle');
INSERT INTO AddGuest1 VALUES
(55111,'Matthew Atif');
INSERT INTO AddGuest1 VALUES
(52111,'Caomh Malone');
INSERT INTO AddGuest1 VALUES
(52322,'Rabia Victoria');
INSERT INTO AddGuest1 VALUES
(52333,'Jo Moodie');
INSERT INTO AddGuest1 VALUES
(53111,'Isiah Chapman');
INSERT INTO AddGuest1 VALUES
(54222,'Omid Rahmani');
INSERT INTO AddGuest1 VALUES
(54555,'Darren Legore');
INSERT INTO AddGuest1 VALUES
(56111,'Cristina Laporte');
INSERT INTO AddGuest1 VALUES
(56222,'Angela Georgiou');
```

```
CREATE TABLE AddGuest2
(AddID NUMBER(5) NOT NULL PRIMARY KEY,
```

```
MainGuestID NUMBER(5),  
FOREIGN KEY(MainGuestID) REFERENCES MainGuest(MainGuestID)  
);
```

```
INSERT INTO AddGuest2 VALUES  
  (42111, 42222);  
INSERT INTO AddGuest2 VALUES  
  (43111,43333);  
INSERT INTO AddGuest2 VALUES  
  (43313,43333);  
INSERT INTO AddGuest2 VALUES  
  (44111,44444);  
INSERT INTO AddGuest2 VALUES  
  (44222,44444);  
INSERT INTO AddGuest2 VALUES  
  (45111,45555);  
INSERT INTO AddGuest2 VALUES  
  (45222,45555);  
INSERT INTO AddGuest2 VALUES  
  (46111,46666);  
INSERT INTO AddGuest2 VALUES  
  (48111,48888);  
INSERT INTO AddGuest2 VALUES  
  (48222,48888);  
INSERT INTO AddGuest2 VALUES  
  (55111,51111);  
INSERT INTO AddGuest2 VALUES  
  (52111,52222);  
INSERT INTO AddGuest2 VALUES  
  (52322,52222);  
INSERT INTO AddGuest2 VALUES  
  (52333,52222);  
INSERT INTO AddGuest2 VALUES  
  (53111,53333);  
INSERT INTO AddGuest2 VALUES  
  (54222,53333);  
INSERT INTO AddGuest2 VALUES  
  (54555,54444);  
INSERT INTO AddGuest2 VALUES  
  (56111,54444);  
INSERT INTO AddGuest2 VALUES  
  (56222,54444);
```

```
CREATE TABLE Rooms  
  (RoomID NUMBER(3) NOT NULL PRIMARY KEY,  
   Types VARCHAR2(12),  
   MainGuestID NUMBER(5),
```

FOREIGN KEY(MainGuestID) REFERENCES MainGuest(MainGuestID)
);

```
INSERT INTO Rooms VALUES
(111,'Excellent',41111);
INSERT INTO Rooms VALUES
(122,'Deluxe',42222);
INSERT INTO Rooms VALUES
(133,'Magnificent',43333);
INSERT INTO Rooms VALUES
(144,'Excellent',44444);
INSERT INTO Rooms VALUES
(155,'Magnificent',45555);
INSERT INTO Rooms VALUES
(166,'Magnificent',46666);
INSERT INTO Rooms VALUES
(177,'Deluxe',47777);
INSERT INTO Rooms VALUES
(188,'Deluxe',48888);
INSERT INTO Rooms VALUES
(199,'Magnificent',49999);
INSERT INTO Rooms VALUES
(129,'Excellent',51111);
INSERT INTO Rooms VALUES
(191,'Magnificent',52222);
INSERT INTO Rooms VALUES
(119,'Excellent',53333);
INSERT INTO Rooms VALUES
(118,'Magnificent',54444);
```

```
CREATE TABLE RoomsType
(Types VARCHAR2(13),
BasicPrice NUMBER(5)
);
```

```
INSERT INTO RoomsType VALUES
('Excellent',1000);
INSERT INTO RoomsType VALUES
('Deluxe',2000);
INSERT INTO RoomsType VALUES
('Magnificent',3000);
INSERT INTO RoomsType VALUES
('Excellent',1000);
INSERT INTO RoomsType VALUES
('Magnificent',3000);
INSERT INTO RoomsType VALUES
('Magnificent',3000);
INSERT INTO RoomsType VALUES
('Deluxe',2000);
```



```

INSERT INTO RoomsType VALUES
    ('Deluxe',2000);
INSERT INTO RoomsType VALUES
    ('Magnificent',3000);
INSERT INTO RoomsType VALUES
    ('Excellent',1000);
INSERT INTO RoomsType VALUES
    ('Magnificent',3000);
INSERT INTO RoomsType VALUES
    ('Excellent',1000);
INSERT INTO RoomsType VALUES
    ('Magnificent',3000);

```

```

CREATE TABLE Booking
    (BookingReference NUMBER(10) NOT NULL PRIMARY KEY,
     StartDate DATE,
     MainGuestID NUMBER(5),
     FOREIGN KEY(MainGuestID) REFERENCES MainGuest(MainGuestID)
    );

```

```

INSERT INTO Booking VALUES
    (11141111,TO_DATE('15-FEB-2018','DD-MON-YYYY'),41111);
INSERT INTO Booking VALUES
    (12242222,TO_DATE('21-JUN-2018','DD-MON-YYYY'),42222);
INSERT INTO Booking VALUES
    (13343333,TO_DATE('7-DEC-2018','DD-MON-YYYY'),43333);
INSERT INTO Booking VALUES
    (14444444,TO_DATE('5-JUN-2018','DD-MON-YYYY'),44444);
INSERT INTO Booking VALUES
    (15545555,TO_DATE('4-SEPT-2018','DD-MON-YYYY'),45555);
INSERT INTO Booking VALUES
    (16646666,TO_DATE('14-FEB-2018','DD-MON-YYYY'),46666);
INSERT INTO Booking VALUES
    (17747777,TO_DATE('25-JUN-2018','DD-MON-YYYY'),47777);
INSERT INTO Booking VALUES
    (18848888,TO_DATE('4-OCT-2018','DD-MON-YYYY'),48888);
INSERT INTO Booking VALUES
    (19949999,TO_DATE('14-JUL-2018','DD-MON-YYYY'),49999);
INSERT INTO Booking VALUES
    (12951111,TO_DATE('23-SEPT-2018','DD-MON-YYYY'),51111);
INSERT INTO Booking VALUES
    (19152222,TO_DATE('26-MAR-2018','DD-MON-YYYY'),52222);
INSERT INTO Booking VALUES
    (11953333,TO_DATE('23-AUG-2018','DD-MON-YYYY'),53333);
INSERT INTO Booking VALUES
    (11854444,TO_DATE('13-SEPT-2018','DD-MON-YYYY'),54444);

```

```

CREATE TABLE FinalPrice
    (BookingReference NUMBER(10) NOT NULL PRIMARY KEY,

```

```
Price NUMBER(7),
NoNights NUMBER(3)
);
```

```
INSERT INTO FinalPrice VALUES
(11141111,7000,7);
INSERT INTO FinalPrice VALUES
(12242222,1100,5);
INSERT INTO FinalPrice VALUES
(13343333, 4320,12);
INSERT INTO FinalPrice VALUES
(14444444,4300,4);
INSERT INTO FinalPrice VALUES
(15545555,5000,5);
INSERT INTO FinalPrice VALUES
(16646666,2300,2);
INSERT INTO FinalPrice VALUES
(17747777,8500,8);
INSERT INTO FinalPrice VALUES
(18848888, 6500,6);
INSERT INTO FinalPrice VALUES
(19949999, 1750,1);
INSERT INTO FinalPrice VALUES
(12951111, 9000,9);
INSERT INTO FinalPrice VALUES
(19152222, 1000,1);
INSERT INTO FinalPrice VALUES
(11953333, 1320,1);
INSERT INTO FinalPrice VALUES
(11854444, 3000,3);
```

```
CREATE TABLE RoomBooking
(BookingReference NUMBER(10) NOT NULL PRIMARY KEY,
RoomID NUMBER(3),
MainGuestID NUMBER(5),
FOREIGN KEY(MainGuestID) REFERENCES MainGuest(MainGuestID),
FOREIGN KEY(RoomID) REFERENCES Rooms(RoomID)
);
```

```
INSERT INTO RoomBooking VALUES
(11141111,111,41111);
INSERT INTO RoomBooking VALUES
(12242222,122, 42222);
INSERT INTO RoomBooking VALUES
(13343333, 133,43333);
INSERT INTO RoomBooking VALUES
(14444444,144, 44444);
```

```
INSERT INTO RoomBooking VALUES
    (15545555,155,45555);
INSERT INTO RoomBooking VALUES
    (16646666,166,46666);
INSERT INTO RoomBooking VALUES
    (17747777,177,47777);
INSERT INTO RoomBooking VALUES
    (18848888, 188,48888);
INSERT INTO RoomBooking VALUES
    (19949999, 199,49999);
INSERT INTO RoomBooking VALUES
    (12951111, 129,51111);
INSERT INTO RoomBooking VALUES
    (19152222, 191,52222);
INSERT INTO RoomBooking VALUES
    (11953333, 119,53333);
INSERT INTO RoomBooking VALUES
    (11854444, 118,54444);
```

```
COMMIT;
```

```
SET TERMOUT ON
PROMPT Demonstration table build is complete.
```

4. Set of “create view” commands with explanations of their usage (All view relations shows are a portion of the total data)

4.1 First view

```
SQL> create view addit AS
2 select addguest1.addid, addname, mainguestid
3 from addguest1 inner join addguest2 on
4 addguest1.addid = addguest2.addid;
```

```
SELECT *
2 FROM ADDIT;
```

ADDID	ADDNAME	MAINGUESTID
42111	Anthony Phillips	42222
43111	Rachel Green	43333
43313	Sarah White	43333
44111	Atif Ali	44444

In our normalised relations we could not access the names of the additional guests for each main guest to answer question 2. This is because the AddGuest1 relation did not have MainGuestID to join MainGuest with AddGuest1 . Therefore we created a view between AddGuest1 and AddGuest2.

4.2 Second view

```
CREATE VIEW VIEWTABLE1 AS
2 SELECT MAINGUESTID,PRICE,ROOMID
3 FROM FINALPRICE INNER JOIN ROOMBOOKING ON
4 FINALPRICE.BOOKINGREFERENCE = ROOMBOOKING.BOOKINGREFERENCE;
```

View created.

MAINGUESTID	PRICE	ROOMID
41111	7000	111
42222	1100	122
43333	5000	133
44444	4300	144

In this case the 'Price' attribute could not be accessed from the FinalPrice relation or joined with MainGuest to answer question 1. Therefore we made a virtual relation by using inner join using the BookingReference primary key between FinalPrice and RoomBooking. We made the BookingReference primary keys equal to each other in the command to maintain referential integrity.

4.3 Third view

```
CREATE VIEW VIEWTABLE3 AS
2 SELECT BOOKINGREFERENCE, ROOMS.MAINGUESTID, TYPES
3 FROM ROOMS INNER JOIN ROOMBOOKING ON
4 ROOMS.ROOMID = ROOMBOOKING.ROOMID;
```

BOOKINGREFERENCE MAINGUESTID TYPES

11141111	41111	Excellent
13343333	43333	Magnificent
14444444	44444	Excellent
15545555	45555	Magnificent

We created the view to combine the Rooms relation and RoomBooking to access the room 'type' to answer question 11. In this case the primary key 'RoomId' of Rooms and foreign key 'RoomID' of RoomBooking is used to connect the two relation to form the virtual table.

5. The set of 11 SQL queries with listings they output when running them.

QUERY 1

This query is used to list the main guest names that are paying between 1500 and 5000.

```
SELECT NAME, PRICE
2 FROM VIEWTABLE1 INNER JOIN MAINGUEST ON
3 VIEWTABLE1.MAINGUESTID = MAINGUEST.MAINGUESTID
4 WHERE PRICE BETWEEN 1500 AND 5000;
```

NAME	PRICE

Ross Geller	5000
Sara Chamberlain	4300
Theodore Russel	5000
Kyle Benoit	2300
Tazim Chowdhury	1750
Tony Trump	3000

6 rows selected.

QUERY 2

This query is used to list the main guest with their additional guests names as well as the main guest ID.

```
SQL> select mainguest.mainguestid, name, addname
2 from addit inner join mainguest on
3 mainguest.mainguestid = addit.mainguestid;
```

MAINGUESTID	NAME	ADDNAME

43333	Ross Geller	Sarah White
43333	Ross Geller	Rachel Green
44444	Sara Chamberlain	Shaheen Dhokia
44444	Sara Chamberlain	Atif Ali
45555	Theodore Russel	Ekram Rizvi
45555	Theodore Russel	Chris Feynman
46666	Kyle Benoit	Sajida Rose
48888	Athena Scourta	Pauline Kyle
48888	Athena Scourta	Demitri Iacovou
51111	Cynthia Garrick	Matthew Atif
52222	Sophie Smith	Jo Moodie

MAINGUESTID	NAME	ADDNAME
-------------	------	---------

```

-----
52222    Sophie Smith    Rabia Victoria
52222    Sophie Smith    Caomh Malone
53333    Pauline Duncan   Omid Rahmani
53333    Pauline Duncan   Isiah Chapman
54444    Tony Trump      Angela Georgiou
54444    Tony Trump      Cristina Laporte
54444    Tony Trump      Darren Legore
42222    Mark Seth       Anthony Phillips

```

19 rows selected.

SQL>

QUERY 3

This query is used to get the average price paid by the guests during September 2018.

```

SELECT AVG(PRICE)
2 FROM BOOKING INNER JOIN FINALPRICE ON
3 BOOKING.BOOKINGREFERENCE = FINALPRICE.BOOKINGREFERENCE
4 WHERE STARTDATE BETWEEN '01-SEP-2018' AND '30-SEP-2018';

```

AVG(PRICE)

```

-----
5666.66667

```

QUERY 4

This query is used to count the number of additional guests that stayed in magnificent type of room.

```

SELECT COUNT(ADDID)
2 FROM ROOMS INNER JOIN ADDGUEST2 ON
3 ROOMS.MAINGUESTID = ADDGUEST2.MAINGUESTID
4 WHERE TYPES ='Magnificent';

```

COUNT(ADDID)

```

-----
11

```

QUERY 5

This query is used to show the maximum price paid and name of main guests that paid it

```
SELECT NAME, PRICE
2 FROM MAINGUEST
3 INNER JOIN VIEWTABLE1 ON MAINGUEST.MAINGUESTID =
VIEWTABLE1.MAINGUESTID
4 WHERE VIEWTABLE1.PRICE = (SELECT MAX(PRICE)
5 FROM VIEWTABLE1);
```

NAME	PRICE

Cynthia Garrick	9000

QUERY 6

This query is used to show the names and the booking references of main guests who booked a room between 01 June 2018 to 31 August 2018.

```
SELECT BOOKINGREFERENCE, NAME
2 FROM MAINGUEST INNER JOIN BOOKING ON
3 MAINGUEST.MAINGUESTID = BOOKING.MAINGUESTID
4 WHERE STARTDATE BETWEEN '01-JUN-2018' AND '31-AUG-2018';
```

BOOKINGREFERENCE	NAME

14444444	Sara Chamberlain
17747777	Matthew Johnson
19949999	Tazim Chowdhury
11953333	Pauline Duncan
12242222	Mark Seth

QUERY 7

This query is used to count the number of guests that booked a room between 1 June 2018 and 31 August 2018 without listing their details.

```
SELECT COUNT (NAME)
2 FROM MAINGUEST
3 INNER JOIN BOOKING ON MAINGUEST.MAINGUESTID= BOOKING.MAINGUESTID
4 WHERE BOOKING.STARTDATE BETWEEN '1-JUN-2018' AND '31-AUG-2018';
```

COUNT(NAME)

5

QUERY 8

This query is used to find all the main guest names that start with “T”.

```
SELECT NAME
2 FROM MAINGUEST
3 WHERE NAME LIKE 'T%';
```

NAME

Theodore Russel
Tazim Chowdhury
Tony Trump

QUERY 9

This query is used to find the average final price and group by the number of additional guests only if the average is greater than 1500. Show the results in descending order.

```
SELECT AVG(PRICE), NUMBERGUESTS
2 FROM VIEWTABLE1
3 INNER JOIN MAINGUEST ON MAINGUEST.MAINGUESTID =
4 VIEWTABLE1.MAINGUESTID
5 GROUP BY NUMBERGUESTS
6 HAVING AVG(PRICE) > 1500
7 ORDER BY AVG(PRICE) DESC;
```

AVG(PRICE) NUMBERGUESTS

5750 0
4760 2
4133.33333 1

QUERY 10

This query is used to list the names of people who paid the same amount as Ross Geller.

```
SQL> SELECT NAME
2 FROM VIEWTABLE1 INNER JOIN MAINGUEST ON
3 VIEWTABLE1.MAINGUESTID = MAINGUEST.MAINGUESTID
4 WHERE VIEWTABLE1.PRICE = (SELECT PRICE
5 FROM VIEWTABLE1 INNER JOIN MAINGUEST ON
6 VIEWTABLE1.MAINGUESTID = MAINGUEST.MAINGUESTID
7 WHERE NAME = 'Ross Geller');
```

NAME

Ross Geller
Theodore Russel

QUERY 11

This query is used to list the main guests names, their booking reference and main guest ID who have booked the same type of room as John Smith.

```
SELECT MAINGUEST.NAME, VIEWTABLE3.BOOKINGREFERENCE,  
MAINGUEST.MAINGUESTID  
2 FROM VIEWTABLE3 INNER JOIN MAINGUEST ON  
3 VIEWTABLE3.MAINGUESTID = MAINGUEST.MAINGUESTID  
4 WHERE VIEWTABLE3.TYPES = ( SELECT TYPES  
5 FROM ROOMS INNER JOIN MAINGUEST ON  
6 ROOMS.MAINGUESTID = MAINGUEST.MAINGUESTID  
7 WHERE NAME = 'John Smith');
```

NAME	BOOKINGREFERENCE	MAINGUESTID

John Smith	11141111	41111
Sara Chamberlain	14444444	44444
Cynthia Garrick	12951111	51111
Pauline Duncan	11953333	53333