

1) True or False, these Relational States are functionally equivalent?

a) T True or False? (2 pts)

SocialMedia	Characters	Image
Twitter	280	Yes
Snapchat	1196	No
Snapchat	80	Yes
Instagram	2200	Yes

SocialMedia	Characters	Image
Snapchat	1196	No
Instagram	2200	Yes
Snapchat	80	Yes
Twitter	280	Yes

b) F True or False? (2 pts)

SocialMedia	Characters	Image
Twitter	Yes	280
Snapchat	1196	No
Snapchat	80	Yes
2200	Instagram	Yes

SocialMedia	Characters	Image
1196	Snapchat	No
Instagram	2200	Yes
Snapchat	Yes	80
Twitter	280	Yes

c) T True or False? (2 pts)

SocialMedia	Image	Characters
Twitter	Yes	280
Snapchat	No	1196
Snapchat	Yes	80
Instagram	Yes	2,200

SocialMedia	Characters	Image
Snapchat	1196	No
Instagram	2,200	Yes
Snapchat	80	Yes
Twitter	280	Yes

- 2) Are these relations valid? If not, what constraint or constraints do they violate?
 Note, an underlined Attribute Name signifies the Attribute is the Primary Key, or part of the Primary Key.

- a) Not Valid Valid or Not Valid? (2 pts)

If Not Valid, what constraint or constraints were violated?

Key Constraint – The primary key(s) do not uniquely identify each row in the table of the database.

<u>SOCIAL_MEDIA</u>	CHARACTERS	IMAGE
Snapchat	1196	No
Instagram	2,200	Yes
Snapchat	80	Yes
Twitter	280	Yes

- b) Valid Valid or Not Valid? (2 pts)

If Not Valid, what constraint or constraints were violated?

<u>FIRST_NAME</u>	<u>LAST_NAME</u>	OCCUPATION
Reed	Richards	Inventor
Sue	Richards	Actor
Johnny	Storm	NULL
Ben	Grimm	Pilot

- c) Not Valid Valid or Not Valid? (2 pts)

If Not Valid, what constraint or constraints were violated?

Key Constraint – The primary key must be unique and not NULL.

<u>LOCATION</u>	INCIDENT
600 Block of 12th Street	Car Accident
NULL	Missing Bike
200 Block of Pine Street	Jay Walking
1000 Block of Forum Avenue	Car Accident

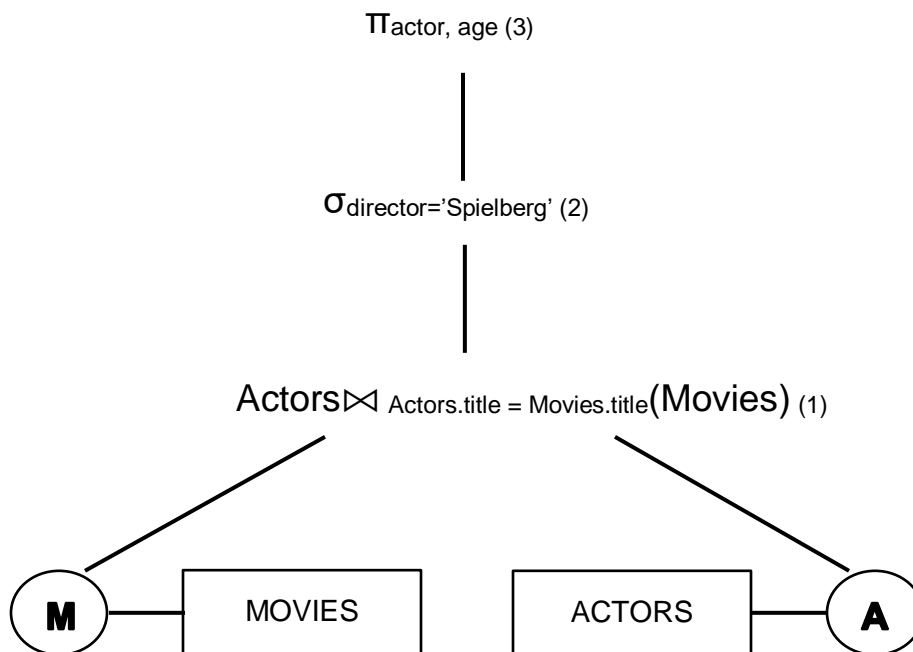
d) Valid Valid or Not Valid? (2 pts)

If Not Valid, what constraint or constraints were violated?

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
1011	Barbara	Smith
1015	Barbara	Henson
1001	Barbara	Duvall
2112	Dale	Young

3) Write the Query Tree for the following relational algebra expression (5 pts).

$\pi_{\text{actor, age}}(\sigma_{\text{director}='Spielberg'}(\text{Actors} \bowtie_{\text{Actors.title} = \text{Movies.title}}(\text{Movies})))$



Use the following table, ART, to answer all parts of question 4, 5, and 6.

ART

ITEM	NAME	ARTIST	ORIGIN_ID	DATING	MEDIA
9182	One Hundred Horses	Lang Shining	1111	960 to 1127	Painting
6922	The Great Wave off Kanagawa	Katsushika Hokusai	6943	1829 to 1833	Painting
2049	Toluvila statue	NULL	8415	300 to 400	Statuary
2038	Sasanian silver vessel	NULL	1598	700 to 722	Silver
3964	Nymph of the Luo River	Gu Kaizhi	1111	317 to 420	Painting
3097	The Hunt of the Unicorn	NULL	3543	1680	Tapestries

ORIGIN

<u>ORIGIN_ID</u>	LOCATION
1111	China
3543	France
6943	Japan
8415	Sri Lanka
1598	Tajikistan

- 4) Use the above tables ART and ORIGIN to answer parts a, b, c, d, and e.
- Give an example of three valid Superkeys for the ART table (3 pts).
 - ITEM
 - ITEM, NAME
 - NAME
 - Give an example of two Candidate keys for the ART table (2 pts).
 - ITEM
 - NAME
 - State a Primary key and a Unique key for the ART table and explain why you chose the candidate key you did (2 pts).

Primary Key: ITEM ; Unique Key: NAME

These were chosen because they meet the requirements of a minimal super key with no redundant attributes.

- d. Describe a specific example of an Insert operation into ART that will violate your Primary Key constraint for ART (2 pts).
Insert<'9182', 'Mona Lisa', 'Leonardi Da Vinci', '1234', '1503', 'Painting'> into ART.
Result: This insertion violates the primary key constraint because another tuple already exists in the ART relation with that ITEM, and so it is rejected.
- e. Describe a specific example of an Update operation into ART that will violate a Domain constraint for an Attribute of ART and state the assumed Domain constraint of the Attribute (3 pts).
Assumption: MEDIA must be a varchar value with <= 15 characters.
UPDATE the media of the ART tuple with the ITEM = '9182' to 'Oil Painting on Canvas'
Result: This update violates the domain constraint because the assumed value of MEDIA can only be less than or equal to 15 characters, and so it is rejected.
5. Use the above tables ART and ORIGIN to answer parts a, b, c, and d.
Note, ORIGIN_ID in the table ART is a Foreign Key in table ORIGIN that references the ART table.
- a. Describe a specific example of an Insert operation that will violate Referential Integrity constraint between ART and ORIGIN (2 pts).
Insert<'9182', 'Mona Lisa', 'Leonardi Da Vinci', '1234', '1503', 'Painting'> into ART.
Result: This insertion violates the referential integrity constraint between ART and ORIGIN because there is no corresponding tuple in ORIGIN with ORIGIN_ID = 1234, and so it is rejected.
- b. Describe a specific example of an Update operation on ORIGIN that will violate the Key constraints of the ORIGIN table (2 pts).
UPDATE the ORIGIN_ID in the ORIGIN tuple with ORIGIN_ID = '1111' to '1598'.
Result: This update operation violates the key constraints of the ORIGIN relation because another tuple already exists in ORIGIN with that ORIGIN_ID, and so it is rejected.

- c. Describe the step or steps necessary to Insert the following item into the ART table (3 pts): (Guernica, Pablo Picasso, France, 1937, Painting)

In order to insert the following into the table, we need to verify the ORIGIN relation contains France. If it does, return the ORIGIN_ID corresponding to the LOCATION of France. If it does not, a new LOCATION/ORIGIN_ID tuple containing France must be added to the ORIGIN table. Finally, an ITEM attribute must be assigned to the tuple. The resulting operation would look like this:

Insert<'7878', 'Guernica', 'Pablo Picasso', '3543', '1937', 'Painting'> into ART.

Note: for this operation, 7878 was chosen for the ITEM attribute as a arbitrary, unique/non-redundant value.

- d. Describe the step or steps necessary to Insert the following item into the ART table (4 pts): (Skrik, Edvard Munch, Norway, 1893, Painting)

In order to make the following insertion into the table, we would need to verify the existence of Norway in the ORIGIN relation. Norway currently does not exist in the ORIGIN relation so an insertion of Norway must first be added to the ORIGIN table. Next, an ITEM attribute must be assigned to the tuple. The resulting operations would look like this:

Insert<'1737', 'Norway'> into ORIGIN.

Insert<'8100', 'Skrik', 'Edvard Munch', '1737', '1893', 'Painting'> into ART.

Note: For these operations, 1737 and 8100 were chosen as arbitrary, unique/non-redundant values for ORIGIN_ID and ITEM. The second insertion can only be executed with the assumption that the first insertion is executed successfully.

6. Use the above tables ART and ORIGIN to answer parts a, b, c, and d.

- a. Write the Formal Relational Algebra Expression that would be used to join the table ART and the table ORIGIN together (3 pts).

$$\text{ART_ORG} \leftarrow \text{ART} \bowtie_{\text{ORIGIN_ID}=\text{ORIGIN_ID}} \text{ORIGIN}$$

- b. Write the Informal SQL statement that would be used to join ART and ORIGIN together (3 pts).

```
SELECT *
FROM ART
INNER JOIN ORIGIN on ART.ORIGIN_ID=ORIGIN.ORIGIN_ID
```

- c. Write the Informal Select statement you would use to list all the Artworks and the data on them in the database from France (3 pts).

```
SELECT *  
FROM ART  
INNER JOIN (SELECT ORIGIN_ID  
FROM ORIGIN  
WHERE LOCATION='France') FRANCE_ORIGIN ON ART.ORIGIN_ID =  
FRANCE_ORIGIN.ORIGIN_ID
```

- d. Write the Informal statement you would use to list all of the Artworks and the creating Artists (3 pts).

```
SELECT DISTINCT ART.NAME, ART.ARTIST  
FROM ART
```

7. Connect to the scsp database and use the os_hr tables to answer the following questions.

KOOBP.OS_HR_JOBS
 KOOBP.OS_HR_JOB_HISTORY
 KOOBP.OS_HR_DEPARTMENTS
 KOOBP.OS_HR_EMPLOYEES
 KOOBP.OS_HR_LOCATIONS
 KOOBP.OS_HR_REGIONS
 KOOBP.OS_HR_COUNTRIES

For each answer, show the SQL statement you used to get it and results.

- a. Get an employee contact list that lists First Name, Last Name, Email Address, and Phone Number for each employee (3 pts).

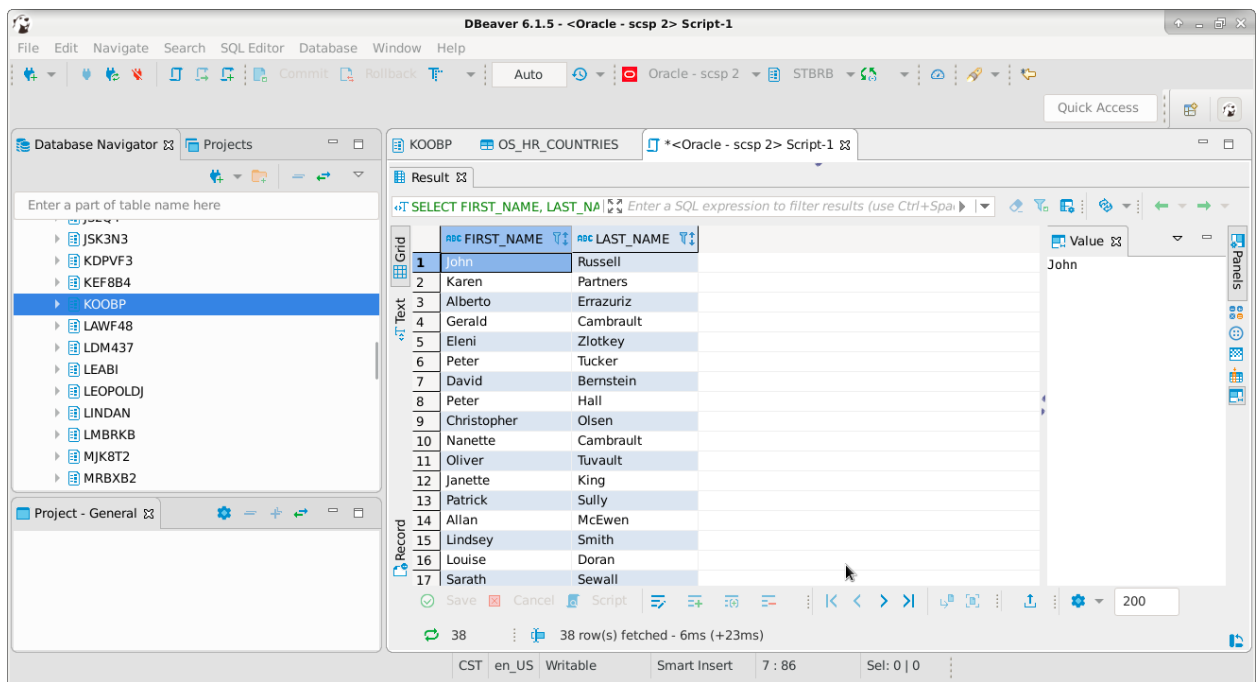
SELECT FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER
 FROM KOOBP.OS_HR_EMPLOYEES;

The screenshot shows the DBeaver 6.1.5 interface. The 'Database Navigator' on the left shows the 'KOOBP' database selected. The 'SQL Editor' in the center contains the query: `SELECT FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER FROM KOOBP.OS_HR_EMPLOYEES;`. The 'Result' tab on the right displays the query results in a grid format. The status bar at the bottom indicates '107 row(s) fetched - 5ms (+124ms)'.

	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	Steven	King	SKING	515.123.4567
2	Neena	Kochhar	NKOCHHAR	515.123.4568
3	Lex	De Haan	LDEHAAN	515.123.4569
4	Alexander	Hunold	AHUNOLD	590.423.4567
5	Bruce	Ernst	BERNST	590.423.4568
6	David	Austin	DAUSTIN	590.423.4569
7	Valli	Pataballa	VPATABAL	590.423.4560
8	Diana	Lorentz	DLORENTZ	590.423.5567
9	Nancy	Greenberg	NGREENBE	515.124.4569
10	Daniel	Faviet	DFAVIET	515.124.4169
11	John	Chen	JCHEN	515.124.4269
12	Ismael	Sciarra	ISCIARRA	515.124.4369
13	Jose Manuel	Urman	JMURMAN	515.124.4469
14	Luis	Popp	LPOPP	515.124.4567
15	Den	Raphaely	DRAPHEAL	515.127.4561
16	Alexander	Khoo	AKHOO	515.127.4562
17	Shelli	Baida	SBAIDA	515.127.4563

- b. Get a list for employees, First Name and Last Name, who are not in the United States. (5 pts)

```
SELECT FIRST_NAME, LAST_NAME
FROM KOOBP.OS_HR_EMPLOYEES
INNER JOIN (
SELECT DEPARTMENT_ID
FROM KOOBP.OS_HR_DEPARTMENTS
INNER JOIN (SELECT LOCATION_ID FROM KOOBP.OS_HR_LOCATIONS
WHERE COUNTRY_ID != 'US') NON_US_LOCATIONS ON
OS_HR_DEPARTMENTS.LOCATION_ID =
NON_US_LOCATIONS.LOCATION_ID
) NON_US_EMPLOYEES ON OS_HR_EMPLOYEES.DEPARTMENT_ID =
NON_US_EMPLOYEES.DEPARTMENT_ID;
```



- c. Get a list of managers, manager_id, and the count of employees who report to them (5 pts).

```
SELECT FIRST_NAME, LAST_NAME, EMP_COUNT
FROM KOOBP.OS_HR_EMPLOYEES EMPLOYEES
RIGHT JOIN (
SELECT MANAGER_ID, COUNT(EMPLOYEE_ID) AS EMP_COUNT
FROM KOOBP.OS_HR_EMPLOYEES
GROUP BY MANAGER_ID) MANAGER_LIST
ON EMPLOYEES.EMPLOYEE_ID = MANAGER_LIST.MANAGER_ID;
```

The screenshot shows the DBeaver 6.1.5 interface with the following components:

- Database Navigator:** Lists database schemas including JSK3N3, KDPVF3, KEF8B4, KOOBP (selected), LAW48, LDM437, LEABI, LEOPOLDJ, LINDAN, LMBRKB, MRXB2, MY7T4, NJL44F, and NLL9Q2.
- SQL Editor:** Contains the SQL script for finding managers and their employee counts.
- Result Grid:** Displays the query results in a table with columns FIRST_NAME, LAST_NAME, and EMP_COUNT. The results are as follows:

FIRST_NAME	LAST_NAME	EMP_COUNT
Steven	King	14
Neena	Kochhar	5
Lex	De Haan	1
Alexander	Hunold	4
Nancy	Greenberg	5
Den	Raphaely	5
Matthew	Weiss	8
Adam	Fripp	8
Payam	Kaufling	8
Shanta	Vollman	8
Kevin	Mourgos	8
John	Russell	6
Karen	Partners	6
- Status Bar:** Indicates "18 row(s) fetched - 1ms (+15ms)".