



KANDIDAT

10944

PRØVE

TDT4145 1 Datamodellering og databasesystemer

Emnekode	TDT4145
Vurderingsform	Semesterprøve
Starttid	24.02.2026 09:00
Sluttid	24.02.2026 11:00
Sensurfrist	17.03.2026 23:59
PDF opprettet	24.02.2026 10:59

Section 1

Oppgave	Tittel	Oppgavetype
i	Frontpage	Informasjon eller ressurser
1	Q1.1 Relational Operators (5%)	Flervalg
2	Q1.2 Keys (5%)	Flervalg
3	Q1.3 Join Types (5%)	Flervalg (flere svar)
4	Q1.4 RA Result (5%)	Fyll inn tall
5	Q1.5 SQL Query Result 1 (5%)	Fyll inn tall
6	Q1.6 SQL Query Result 2 (5%)	Fyll inn tall
7	Q1.7 SQL and Relational Algebra (5%)	Flervalg
8	Q1.8 Terminology (5%)	Flervalg
9	Q1.9 SQL Query Processing (5%)	Flervalg
10	Q1.10 Transactions (5%)	Nedtrekk
11	Q1.11 Foreign keys (5%)	Flervalg
12	Q1.12 ER to Relations (5%)	Flervalg
13	Q1.13 Constraints in ER (5%)	Flervalg (flere svar)
14	Q1.14 Normal Forms (5%)	Flervalg

Section 2

Oppgave	Tittel	Oppgavetype
15	Q2.1 - SQL DDL (10%)	Programmering
16	Q2.2 SQL Query 1 (5%)	Programmering
17	Q2.3 SQL Query 2 (5%)	Programmering
18	Q2.4 SQL Query 3 (5%)	Programmering

19	Q2.5 SQL Query 4 (5%)	Programmering
20	Q.0 Comments (0%)	Langsvar

1 Q1.1 Relational Operators (5%)

Hvilken av de følgende er ikke en basis/fundamental operator i relasjonsalgebraen?

Velg ett alternativ:

- Set intersection.
- Selection.
- Set union.
- Projection.
- No other alternative is correct.

2 Q1.2 Keys (5%)

Den minimale mengden av en supernøkkel (superkey) kalles for:

Velg ett alternativ:

- Primary key.
- Discriminator.
- Minimal subkey.
- No other alternative is correct.
- Candidate key.

3 Q1.3 Join Types (5%)

Velg alle utsagn fra det følgende som er sanne (**true**).

Velg ett eller flere alternativer

- The result size of a full outer join is always smaller than that of the cross product.
- A full outer join always results in tuples that contain null values.
- The order in which inner joins are written does not affect the result of a query.
- Every natural join operation can be replaced by a combination of the cross product and the selection operator.
- The size of the result of a left outer join is always equal to the size of the left relation.

4 Q1.4 RA Result (5%)

Følgende tabeller er gitt:

employee		
name	salary	manager
Alice	40000	John
Trevor	38000	John
Bob	35000	Trevor
Jim	35000	Trevor
Dorothy	36000	Trevor

person		
f_name	salary	city
Siobhan	40000	Trondheim
Marine	35000	Zurich
Yiyu	35000	Athens

Hvor mangle tupler vil det følgende algebrauttrykket returnere?

$$\Pi_{\text{manager}}(\sigma_{\text{salary} < 40000}(\text{employee} \bowtie \text{person}))$$

Skriv svaret ditt her: .

5 Q1.5 SQL Query Result 1 (5%)

Den følgende tabellen er gitt:

employee

name	salary	manager
John	50000	NULL
Alice	40000	John
Trevor	38000	John
Bob	35000	Trevor
Jim	35000	Trevor
Dorothy	36000	Trevor
Samuel	36000	Alice

Hva er resultatet av det følgende SQL-query?

```
SELECT count(*)
FROM employee AS a, employee AS b
WHERE a.salary < 40000
```

Skriv svaret ditt her:

35

6 Q1.6 SQL Query Result 2 (5%)

Den følgende tabellen er gitt:

employee

name	salary	manager
John	50000	NULL
Alice	40000	John
Trevor	38000	John
Bob	35000	Trevor
Jim	35000	Trevor
Dorothy	36000	Trevor
Samuel	36000	Alice

Hva er resultatet av det følgende SQL-query?

```
SELECT b.salary
FROM employee AS a JOIN employee AS b ON (a.manager = b.name)
WHERE a.salary < 36000
ORDER BY b.salary
LIMIT 1
```

Skriv svaret ditt her:

38000

7 Q1.7 SQL and Relational Algebra (5%)

Det følgende skjema er gitt:

$\text{bar}(\underline{\text{bar_id}}, \text{name}, \text{location})$

$\text{drink}(\underline{\text{drink_id}}, \text{name}, \text{price})$

$\text{serves}(\underline{\text{bar_id}}, \underline{\text{drink_id}})$

hvor $\text{serves}.\text{bar_id}$ refererer $\text{bar}.\text{bar_id}$ og $\text{serves}.\text{drink_id}$ refererer $\text{drink}.\text{drink_id}$. Hvilken av de følgende relasjonslagebra-uttrykkene er ekvivalent med det følgende SQL-query?

```
SELECT bar.name, drink.name
FROM bar, serves, drink
WHERE bar.bar_id = serves.bar_id
AND drink.drink_id = serves.drink_id
AND price > 20
```

Velg ett alternativ:

- $\Pi_{\text{bar.id}, \text{drink.id}}(\sigma_{\text{price} > 20}(\text{drink} \times \text{serves}))$.
- $\Pi_{\text{bar.name}, \text{drink.name}}(\text{bar} \bowtie_{\text{bar.bar_id}=\text{serves.bar_id}} \text{serves} \bowtie_{\text{serves.drink_id}=\text{drink.drink_id}} (\sigma_{\text{price} > 20} \text{drink}))$.
- No other alternative is correct.
- $\Pi_{\text{bar_name}, \text{drink_name}}(\sigma_{\text{price} > 20}(\text{bar} \bowtie \text{serves} \bowtie \text{drink}))$.
- $\Pi_{\text{bar.name}, \text{drink.name}}(\text{bar} \bowtie \text{sigma}_{\text{price} > 20}(\text{serves} \bowtie_{\text{serves.drink_id}=\text{drink.drink_id}} \text{drink}))$.

8 Q1.8 Terminology (5%)

Hvilken term matcher best den følgende beskrivelsen?

"A relation that is not part of the conceptual model but is made visible to a user as a 'virtual relation'"

Velg ett alternativ:

- No other alternative is correct.
- View.
- Domain.
- Assertion.
- Decomposition.

9 Q1.9 SQL Query Processing (5%)

Når et SQL-query (uten næstede subqueries) prosesseres, hvilken av de følgende uttrykkene utføres til slutt?

Velg ett alternativ:

GROUP BY.

HAVING.

SELECT.

WHERE.

FROM.

10 Q1.10 Transactions (5%)

ACID er en mengde egenskaper som gjør at databasetransaksjoner utføres pålitelig selv under feil eller

samtidig (concurrent) aksess. Disse egenskapene er **Atomicity**, **Consistency**,

Integrity

(Indexing, Irreversibility, Isolation, Integrity) , og **Durability**.

11 Q1.11 Foreign keys (5%)

Se på tabellen T som har blitt skapt med den følgende SQL-koden:

```
CREATE TABLE T (
    tid varchar(20),
    attr integer,
    PRIMARY KEY (tid)
)
```

Du prøver så å skape en ny tabell ved å kjøre den følgende SQL-koden:

```
CREATE TABLE RT (
    tid varchar(20),
    attr varchar(20),
    PRIMARY KEY (tid),
    FOREIGN KEY (attr) REFERENCES T
)
```

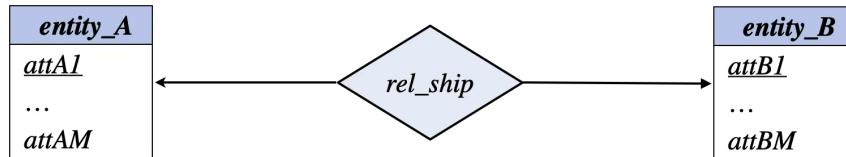
Hvilken av de følgende setningene er **sann (true)**?

Velg ett alternativ:

- The query fails because T.attr and RT.attr have different domains.
- No other alternative is correct.
- The query is executed successfully; RT.attr references T.tid.
- The query fails because no attribute was given in the foreign key definition.
- The query is executed successfully; RT.attr of table RT references T.attr.

12 Q1.12 ER to Relations (5%)

Se på følgende ER-diagram:



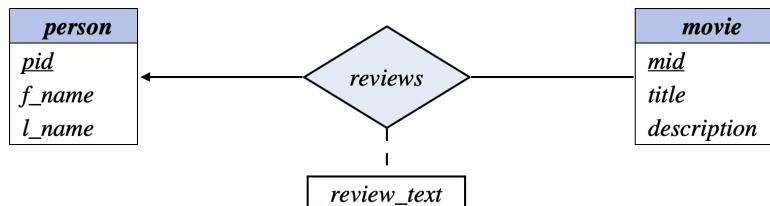
Hva er minimum antall relasjoner (tabeller) som trengs for å mappe dette ER-diagrammet til relasjonsmodellen?

Velg ett alternativ:

- The ER diagram cannot be mapped to relations.
- 1.
- No other alternative is correct.
- 3.
- 2.

13 Q1.13 Constraints in ER (5%)

Se på følgende ER-diagram:



Basert på dette diagrammet, velg alle utsagn fra de følgende som er **sanne (true)**.

Velg ett eller flere alternativer

- No two reviews can have the same review text.
- A person can review multiple movies; a movie can be reviewed by multiple people.
- Each movie must have at least 1 review.
- Each movie can be reviewed by at most one person.
- A person cannot submit two separate reviews for the same movie.

14 Q1.14 Normal Forms (5%)

Alle relasjoner som er på BCNF, er også på:

Velg ett alternativ:

First, second, and third normal form.

First normal form.

Second normal form.

None of the other alternatives.

Third normal form.

15 Q2.1 - SQL DDL (10%)

Vi har oppgitt det følgende relasjonsskjemaet:

```
student(sid, first_name, last_name, email)
quiz(quiz_id, title, quiz_max_points)
question(quiz_id, quest_num, quest_text, quest_max_points)
answer(sid, quiz_id, quest_num, answer_text, points_gained)
```

Skriv SQL-koden som lager tabellen 'question' slik at:

- quiz_id og quest_num er av typen integer.
- quest_max_points er av numeric-typen med maksimalt 2 siffer (digits) og ingen desimaler.
- quest_text er av typen varchar(100).
- De underlinjede attributtene er primærnøkler (primary keys).
- quiz_id er en fremmednøkkel (foreign key) som refererer attributtet med samme navn i tabellen 'quiz'.
- quest_text- og quest_max_points-attributtene kan ikke være NULL.
- Verdien til max_points-attributtet skal være innen [0,20].

Skriv ditt svar her

1	<code>create table question ()</code>	
2	<code>quiz_id int,</code>	
3	<code>quest_num int,</code>	
4	<code>quest_max_points numeric(2, 0) not null,</code>	
5	<code>quest_text varchar(100) not null,</code>	
6	<code>primary key (quiz_id, quest_num),</code>	
7	<code>foreign key (quiz_id) references quiz(quiz_id),</code>	
8	<code>check(quest_max_points >= 0 and quest_max_points <= 20)</code>	
9	<code>)</code>	

16 Q2.2 SQL Query 1 (5%)

Vi har oppgitt det følgende relasjonsskjemaet:

*student(sid, first_name, last_name, email)
quiz(quiz_id, title, quiz_max_points)
question(quiz_id, quest_num, quest_text, quest_max_points)
answer(sid, quiz_id, quest_num, answer_text, points_gained)*

Skriv et SQL-query som finner alle quiz hvor summen av maksimum poeng av alle spørsmålene (questions) overskriver *quiz_max_points til det samme quizet*. Skriv ut *quiz_id* for alle slike quizer.

Skriv ditt svar her

```

1 select qz.quiz_id
2 from quiz qz
3 join question qs on qz.quiz_id = qs.quiz_id
4 group by qz.quiz_id, qz.quiz_max_points
5 having sum(qs.quest_max_points) > qz.quiz_max_points

```

17 Q2.3 SQL Query 2 (5%)

Vi har oppgitt det følgende relasjonsskjemaet:

*student(sid, first_name, last_name, email)
quiz(quiz_id, title, quiz_max_points)
question(quiz_id, quest_num, quest_text, quest_max_points)
answer(sid, quiz_id, quest_num, answer_text, points_gained)*

Skriv et SQL-query som beregner den totale poengsummen per quiz oppnådd av studenten med sid=12345, for quiz hvor tittelen inneholder ordet 'databases'. Skriv ut *quiz_id* og den totale poengsummen oppnådd for dette quizet.

Skriv ditt svar her

```

1 select q.quiz_id, sum(a.points_gained)
2 from quiz q
3 join answer a on q.quiz_id = a.quiz_id
4 where q.title like '%databases%' and a.sid = 12345
5 group by q.quiz_id
6

```

18 Q2.4 SQL Query 3 (5%)

Vi har oppgitt det følgende relasjonsskjemaet:

```
student(sid, first_name, last_name, email)
quiz(quiz_id, title, quiz_max_points)
question(quiz_id, quest_num, quest_text, quest_max_points)
answer(sid, quiz_id, quest_num, answer_text, points_gained)
```

Skriv et SQL-query som skriver ut for quiz med quiz_id=4325 alle quest_text for spørsmål (questions) som har blitt svart for av minst en student. Ikke bruk aggregeringsfunksjoner i svaret ditt.

Skriv ditt svar her

1	select distinct q.quest_text	
2	from question q	
3	join answer a on q.quiz_id = a.quiz_id and q.quest_num = a.quest_num	
4	where q.quiz_id = 4325	

19 Q2.5 SQL Query 4 (5%)

Vi har oppgitt det følgende relasjonsskjemaet:

```
student(sid, first_name, last_name, email)
quiz(quiz_id, title, quiz_max_points)
question(quiz_id, quest_num, quest_text, quest_max_points)
answer(sid, quiz_id, quest_num, answer_text, points_gained)
```

Vi har også oppgitt det følgende SQL-queryet:

```
SELECT first_name, last_name
FROM student
WHERE sid IN (
    SELECT sid
    FROM answer
    WHERE quiz_id = 4326
    AND question_num = 7
    AND points_gained > 10
)
```

Skriv et ekvivalent SQL-query til det over uten å bruke nøstede (nested) queries.

Skriv ditt svar her

1	select distinct s.first_name, s.last_name	
2	from student s	
3	join answer a on a.sid = s.sid	
4	where a.quiz_id = 4326	
5	and a.question_num = 7	
6	and a.points_gained > 10	

20 Q.0 Comments (0%)

Denne "oppgaven" er en mulighet for å informere om omstendigheter som du tenker er **helt nødvendige** å kommunisere til sensor, for at din besvarelse skal bli riktig vurdert. Dette kan for eksempel gjelde antakelser som det var tvingende nødvendig å gjøre.

Du skal ikke bruke dette feltet til å gi *generelle kommentarer* til eksamen, det kan gjøres i Ed eller i e-post til faglærer.

Skriv ditt svar her

Ord: 0