3-timers skriftlig eksamen i "Databaser og Informationssystemer"—DoI, 2021

Datalogisk Institut, Københavns Universitet (DIKU)
Dato: 24. juni, 2021, 9:00–12:00

Forord / Preamble

Solution

Disclaimer: In the following, we present solution sketches for the various questions in the exam. These solution sketches are provided only as a reference, and may lack details that we would expect in a complete answer to the exam. Moreover, some of the questions may admit more than one correct solution, but even in such cases only one solution sketch is provided for brevity. Solution sketches are colored for visibility.

Note, in addition, that the evaluation of the exam takes into account our expectations regarding solutions, the actual formulations provided, the weights of various questions, but also and most importantly the overall evaluation of the exam assignment as a whole. This evaluation is performed by both internal and external examiners and grades are finally provided by discussion and consensus. As such, it is not advised to reason about final grades based on this document.

Denne eksamen bliver evalueret efter 7-trins skalaen med ekstern censur, som angivet i kursusbeskrivelsen. / The exam will be evaluated on the 7-point grading scale with external grading, as announced in the course description.

- Eksamens sprog er dansk eller engelsk. / Your answers must be provided in Danish or English.
- Der er individuel aflevering. Samarbejde eller diskussion af indholdet af denne eksamen med andre studerende eller andre personer er strengt forbudt. / Hand-ins for this exam must be individual, so cooperation with others in preparing a solution is strictly forbidden.
- Det er tilladt at benytte computer samt lærebøger og andet skriftligt materiale benyttet på kurset. Hvis du benytter andre kilder i din besvarelse, skal besvarelsen indeholde kildehenvisninger. / You are allowed to use your computer or other devices, books and other written material from the course. If you use any other sources, they must be cited appropriately.
- Afleveringen består af en enkelt PDF (<50MB) med svar på de stillede spørgsmål. / The submission consists of a single PDF (<50MB) with your answers to the questions asked.

Fejl og uklarheder / Errors and Ambiguities

Hvis du finder fejl og uklarheder/tvetydigheder i eksamenssættet, skal du tydeligt anføre hvilke forudsætninger, som gælder for din besvarelse for det pågældende spørgsmål. Ikke alle spørgsmål har nødvendigvis et korrekt svar, så uklarhed/tvetydighed kan være bevidst. / If you find any errors or ambiguities in the exam text, you should clearly state your assumptions in answering the corresponding questions. Some of the questions may not have a single correct answer, so recall that ambiguities could be intentional.

Forventning til vægtning af spørgsmål / Expectations regarding question weights

Hvert spørgsmål har en angivet indikativ vægtning. Vægtningen benyttes under evalueringen til at prioritere besvarelsen ved karaktergivning. Dog skal det bemærkes at besvarelsen samtidig evalueres som et samlet hele. Med andre ord, den angivne vægtning er kun vejledende for dig ved prioritering afdin tid under eksamen. Du kan ikke forvente, at vægtningen bliver fordelt ligeligt på delspørgsmål. / Questions carry indicative weights. The weights will be used during evaluation to prioritize question answers towards grading; however, recall that the exam is still evaluated as a whole. In other words, we provide weights only as an indication so that you can prioritize your time during the exam, if you need to. You cannot assume that weights will be divided equally among subquestions.

Følgende tabel opsummerer spørgsmålene stillet til denne eksamen med angivelse af vejledendevægte. / The following table summarizes the questions in this exam and their weights.

	Spørgsmål / Question	**Vægtning / Weight**
Q1	E/R model / Entity–Relationship model	17%
Q2	Funktionelle afhængigheder, relationers nøgler, BCNF / Functional dependencies, keys, BCNF	8%
Q3	Relationel calculus / Relational calculus	25%
Q4	Relationel algebra / Relational algebra	25%
Q5	SQL / SQL	25%

1 E/R model / Entity–Relationship model (17%)

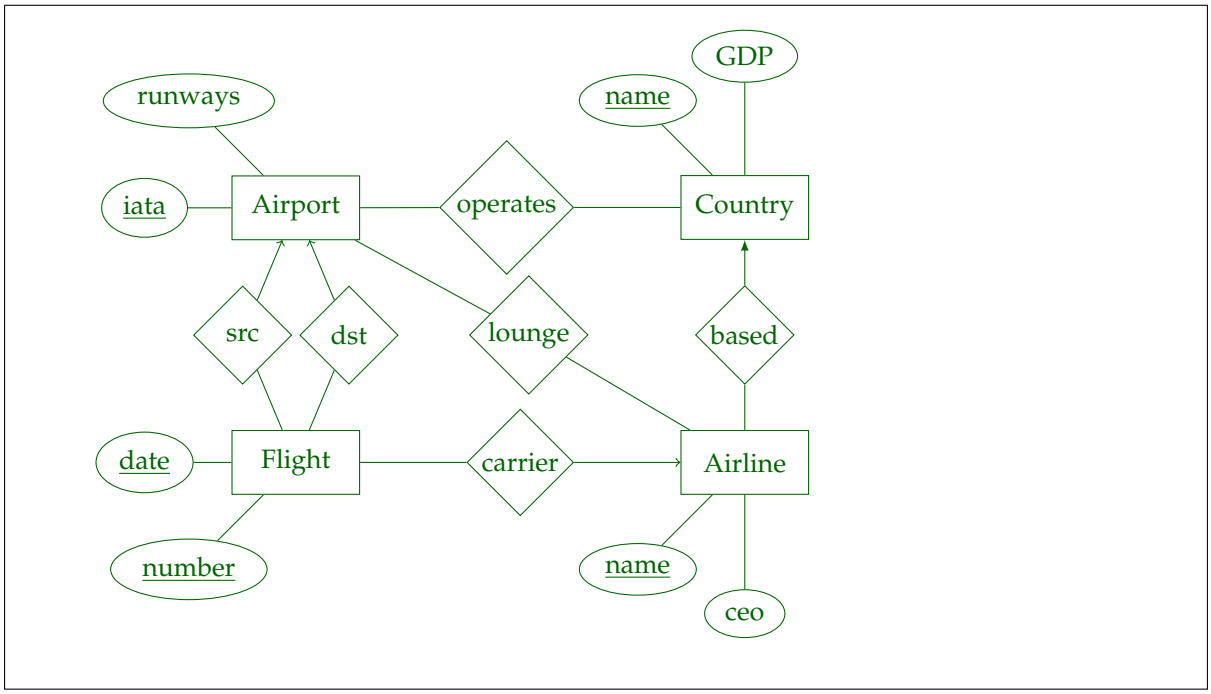
Saliaeri IT Group bruger et relationelt databaseadministrationssystem (DBMS) til at styre deres globale samling af flyoplysninger. Deres udgangspunkt er følgende krav. / The Saliaeri IT Group uses a relational database management system (DBMS) to manage their global collection of flight information. Their starting point are the following requirements.

- Hver flyvning har nøjagtigt et flyselskab. / Every flight has exactly one carrier airline.
- Fly skal have én afgangslufthavn og én destinationslufthavn. / Flights must have one source and one destination airport.
- Lufthavne drives af et eller flere lande. / Airports are operated by one or more countries.
- Flyselskaber er baseret i højst én land. / Airlines are based in at most one country.
- Flyselskaber opretholder lounger i forskellige lufthavne. / Airlines maintain lounges in different airports.

1. Hvilke entiteter og relationer kan du identificere i kravene? For hver entitet skal du opfinde 2–3 meningsfulde attributter. / Which entities and relationships can you identify in the requirements? For each entity, come up with 2–3 meaningful attributes.
2. Formaliser kravene og dine attributter i et E/R-diagram. Understreg nøgler. Skeln tydeligt mellem unikhed (↠) og referentielle integritets (→) constraints. For hver pil argumenter kort (1 sætning), hvorfor du har brugt den ene eller den anden. / Formalize the requirements and your attributes in an E/R diagram. Underline key attributes. Clearly distinguish between uniqueness (↠) and referential integrity (→) constraints. For each arrow briefly argue (1 sentence) why you have used one or the other.
3. Konverter *et* entitetssæt og *et* relationssæt efter eget valg til et relationsdatabaseskema (skriv de to tilsvarende SQL CREATE TABLE-kommandoer). / Convert *one* entity set and *one* relationship set of your choice to a relational database schema (write the two corresponding SQL CREATE TABLE commands).

Solution

1. See diagram.
2. See diagram.
3. ```
   CREATE TABLE airport (iata : VARCHAR PRIMARY KEY, runways : INT)
   CREATE TABLE operates (iata : VARCHAR, name : VARCHAR,
   PRIMARY KEY (iata, name)
   FOREIGN KEY iata REFERENCES airport,
   FOREIGN KEY name REFERENCES country)
   ```
   ```



2 Funktionelle afhængigheder, relationers nøgler, BCNF / Functional dependencies, keys, BCNF (8%)

Givet følgende skema over en relation, samt en tilhørende liste af funktionelle afhængigheder. / Consider the following relation schema with the accompanying list of functional dependencies.

$R(A, B, C)$ $A \to B$ $B \to C$

1. Hvilke af følgende funktionelle afhængigheder kan udledes? Hvorfor / hvorfor ikke (1 sætning for hver forklaring)? / Which of the following functional dependencies can be derived? Why/why not (1 sentence for each explanation)?

   (i) $AB \to A$, (ii) $BC \to A$, and (iii) $A \to ABC$

2. Angiv alle nøgler til R. Forklar kort (1 sætning). / List all of R's keys. Explain briefly (1 sentence).

3. Hvad er alle supernøgler til relation R, der ikke er nøgler? Forklar kort (1 sætning). / What are all the superkeys for relation R that are not keys? Explain briefly (1 sentence).

4. Er relation R i Boyce-Codd normalform (BCNF)? Hvorfor / hvorfor ikke? Hvis ikke, hvordan vil du gå videre for at opnå BCNF? (Én sætning pr. spørgsmål.) / Is relation R in Boyce-Codd Normal Form (BCNF)? Why/why not? If not, how would you proceed to achieve BCNF? (One sentence per question.)

Solution

1. $AB \to A$ is a trivial functional dependency. $A \to ABC$ can be derived using Armstrong's axioms. $BC \to A$ can not be derived because A is not on a right-hand-side of any dependency.
2. A is the only key, because A must be contained in a key and from A one can obtain everything.
3. All proper supersets that contain A: ABC, AB, AC

4. Not BCNF. $B \to C$ violates BCNF, B is not a superkey. Decompose along $B \to C$ to obtain $R1(A, B)$ with $A \to B$ and $R2(B, C)$ with $B \to C$.

3 Relationel calculus (RC) / Relational calculus (RC) (25%)

Overvej databaseskemaet for et flyselskab, der består af tre relationer, hvis skemaer er: / Consider the database schema of an airline that consists of three relations, whose schemas are:

```
Flight(flightNr:string, source:string, destination:string, duration:int, aircraftNr:int)
Aircraft(aircraftNr:int, type:string, speed:float, registrationYear:int, seats:int)
Airport(name:string, openedYear:int)
```

Attributterne `source` og `destination` i `Flight`-relationen henviser til lufthavnenavne i `Airport`-relationen. / The attributes `source` and `destination` in the `Flight` relation refer to airport names in the `Airport` relation.

3.1 Skriv RC forespørgsler / Write RC queries

Udtryk følgende forespørgsler ved hjælp af RC (du er *ikke* forpligtet til at bruge den relationelle algebra normalform her). / Express the following queries using RC (you are *not* required to use the relational algebra normal form here).

1. Hvilke flytyper flyver til (`destination`) eller fra (`source`) København? / Which aircraft types fly to (`destination`) or from (`source`) Copenhagen?
2. Hvilke flytyper er de hurtigste? / Which aircraft types are the fastest?
3. Hvilke lufthavne er ældre end alle fly, der flyver fra (`source`) dem? / Which airports are older than all aircrafts that fly from (`source`) them?

Solution

1. $\exists f, src, dst, d, a, sp, r, s.\ \texttt{Aircraft}(a, t, sp, r, s) \wedge \texttt{Flight}(f, src, dst, d, a) \wedge (src = \text{"Copenhagen"} \vee dst = \text{"Copenhagen"})$
2. $\exists a, sp, r, s.\ \texttt{Aircraft}(a, t, sp, r, s) \wedge \neg(\exists a', t', sp', r', s'.\ \texttt{Aircraft}(a', t', sp', r', s') \wedge sp' > sp)$
3. $\exists o.\ \texttt{Airport}(src, o) \wedge \neg(\exists f, dst, d, a, t, sp, r, s.\ \texttt{Flight}(f, src, dst, d, a) \wedge \texttt{Aircraft}(a, t, sp, r, s) \wedge o \geq r$

3.2 Skriv RC forespørgsler i relationel algebra normalform (RANF) / Write RC queries in relational algebra normal form (RANF)

Udtryk følgende forespørgsler ved hjælp af RC-formler i RANF. / Express the following queries using RC formulas in RANF.

1. Hvilke par af fly har samme type, men forskellige antal pladser. Et par skal være anført kun én gang; angiv f.eks. (i, j), men ikke (j, i). / Which pairs of aircrafts have the same type but different numbers of seats. A pair should be listed only once; e.g., list (i, j) but not (j, i).

2. Hvilke flytyper flyver *ikke* til (`destination`) København? / Which aircraft types do *not* fly to (`destination`) Copenhagen?

Solution

1. $\exists t, sp, sp', r, r', s, s'.\ \texttt{Aircraft}(a, t, sp, r, s) \wedge \texttt{Aircraft}(a', t, sp', r', s') \wedge \neg(s = s') \wedge a \leq a'$
2. $\exists a, sp, r, s.\ \texttt{Aircraft}(a, t, sp, r, s) \wedge \neg(\exists f, src, d.\ \texttt{Flight}(f, src, \text{"Copenhagen"}, d, a))$

3.3 Fortolk RC forespørgsler / Interpret RC queries

Beskriv hvad følgende RC-forespørgsler udtrykker på naturligt sprog. / Describe what the following RC queries express in natural language.

1. $\exists x.\ \texttt{Airport}(a, x) \wedge x < 1925$
2. $\exists i.\ \exists a.\ \texttt{Flight}(f, \text{"Copenhagen"}, \text{"Zurich"}, i, a) \wedge i > 120$
3. $\exists f.\ \exists i.\ \exists a.\ \exists y.\ \exists s.\ \texttt{Flight}(f, \text{"Copenhagen"}, \text{"Zurich"}, i, a) \wedge \texttt{Aircraft}(a, t, sp, y, s) \wedge i > 120$

Solution

1. Which airports were opened before 1925?
2. Which flights from Copenhagen to Zurich take longer than 120 (min)?
3. Which aircraft types fly from Copenhagen to Zurich and take longer than 120 (min.) and what is their speed?

4 Relationel algebra (RA) / Relational algebra (RA) (25%)

Vi betragter det samme databaseskema som i den forrige øvelse. / We consider the same database schema as in the previous exercise.

4.1 RC til RA / RC to RA

Udtryk de tre RC-forespørgsler fra spørgsmål 3.3 i (udvidet) RA. / Express the three RC queries from Question 3.3 in (extended) RA.

Solution

1. $\pi_{\texttt{name}}(\sigma_{\texttt{openedYear}<1925}(\texttt{Airport}))$
2. $\pi_{\texttt{flightNr}}(\sigma_{\texttt{source}=\text{"Copenhagen"},\texttt{destination}=\text{"Zurich"},\texttt{duration}>120}(\texttt{Flight}))$
3. $\pi_{\texttt{type},\texttt{speed}}(\sigma_{\texttt{source}=\text{"Copenhagen"},\texttt{destination}=\text{"Zurich"},\texttt{duration}>120}(\texttt{Flight} \bowtie \texttt{Aircraft}))$

4.2 Skriv RA forespørgsler / Write RA queries

Udtryk følgende forespørgsler ved hjælp af (udvidet) RA. / Express the following queries using (extended) RA.

1. Hvilke flytyper er de hurtigste? / Which aircraft types are the fastest?
2. Hvor mange fly flyver til (`destination`) København? / How many aircrafts fly to (`destination`) Copenhagen?
3. Hvor mange forskellige flytyper flyver til (`destination`) København? / How many different types of aircrafts fly to (`destination`) Copenhagen?
4. For hver flytype, der flyver fra (`source`) København, hvor mange fly af den type findes der (i alt inklusive dem, der ikke flyver fra København)? / For each aircraft type that flies from (`source`) Copenhagen, how many aircrafts of that type exist (in total, including the ones that do not fly from Copenhagen)?

Solution

1. $\pi_{\texttt{type}}(\gamma_{\texttt{MAX(speed)}\to\texttt{speed}}(\texttt{Aircraft}) \bowtie \texttt{Aircraft})$
2. $\gamma_{\texttt{CNT(aircraftNr)}}(\sigma_{\texttt{destination}=\texttt{"Copenhagen"}}(\texttt{Flight}))$
3. $\gamma_{\texttt{CNT(type)}}(\delta(\pi_{\texttt{type}}(\sigma_{\texttt{destination}=\texttt{"Copenhagen"}}(\texttt{Flight}) \bowtie \texttt{Aircraft}))$
4. $\gamma_{\texttt{type,CNT(aircraftNr)}}(\pi_{\texttt{type,aircraftNr}}(\sigma_{\texttt{source}=\texttt{"Copenhagen"}}(\texttt{Flight}) \overset{\texttt{o}}{\bowtie}_{\texttt{R}} \texttt{Aircraft}))$

4.3 Fortolk RA forespørgsler / Interpret RA queries

Beskriv hvad følgende RA-forespørgsler udtrykker på naturligt sprog. / Describe what the following RC queries express in natural language.

1. $\pi_{\texttt{Flight.destination}}(\sigma_{\texttt{Flight.source}=\texttt{Airport.name} \wedge \texttt{Airport.openedYear}=1925}(\texttt{Flight} \times \texttt{Airport}))$
2. $\pi_{2021-\texttt{min}\to\texttt{a}}(\gamma_{\texttt{MIN(registrationYear)}\to\texttt{min}}(\texttt{Flight} \bowtie \texttt{Aircraft})))$
3. $\tau_{\texttt{avg}}(\gamma_{\texttt{Flight.source,Flight.destination,AVG(seats)}\to\texttt{avg}}(\texttt{Flight} \bowtie \texttt{Aircraft})))$

Solution

1. Where do aircrafts starting at airports that opened in 1925 fly to?
2. How old is the oldest flying aircraft?
3. What is the average number of seats on flights between every source and destination? Sort the result by the average (ascending).

5 SQL / SQL (25%)

Vi betragter det samme databaseskema som i den forrige øvelse. / We consider the same database schema as in the previous exercise.

5.1 RA til SQL / RA to SQL

Udtryk de tre RA-forespørgsler fra spørgsmål 4.3 i SQL. / Express the three RA queries from Question 4.3 in SQL.

Solution

1. ```
   SELECT destination
   FROM Flight F, Airport A
   WHERE F.source = A.name AND A.openedYear=1925
   ```

2. ```
   SELECT 2021 - MIN(A.registrationYear) AS a
   FROM Flight F NATURAL JOIN Aircraft A
   ```

3. ```
   SELECT F.source, F.destination, AVG(A.seats) AS avg
   FROM Flight F NATURAL JOIN Aircraft A
   GROUP BY F.source, F.destination
   ORDER BY avg ASC
   ```

5.2 Skriv SQL forespørgsler / Write SQL queries

Udtryk følgende forespørgsler ved hjælp af SQL. / Express the following queries using SQL.

1. Hvilke flytyper har det største antal pladser? / Which aircraft types have the largest number of seats?
2. Hvilke fly flyver ikke til (`destination`) København eller blev registreret før 2000. / Which aircrafts do not fly to (`destination`) Copenhagen or were registered before 2000.
3. For hver flyvning beregnes den mindste tilbagelagte afstand. Bemærk, at flere fly kan flyve under det samme flynummer. Du kan antager, at varigheden er angivet i minutter og flyets (konstante) hastighed i kilometer i minuttet. / For each flight compute the minimum covered distance. Note that multiple aircrafts may fly under the same flight number. You may assume that durations are given in minutes and the aircraft's (constant) speed in kilometers per minute.

Solution

1. ```
   SELECT type
   FROM Aircraft
   WHERE seats = (SELECT MAX(seats) FROM Aircraft)
   ```

2. ```
   SELECT aircraftNr
   FROM Aircraft
   WHERE registrationYear < 2000 OR aircraftNr NOT IN
     (SELECT aircraftNr FROM Flights WHERE destination = "Copenhagen")
   ```

3. ```
   SELECT F.flightNr, MIN (F.duration * A.speed) AS distance
   FROM Flight F NATURAL JOIN Aircraft A
   GROUP BY F.flightNr
   ```

5.3 Ændr databasen / Modify the database

Udtryk følgende ændringer i databasen ved hjælp af SQL. / Express the following modifications to the underlying tables using SQL.

1. Slet alle flyvninger, der bruger fly, der blev registreret før 1975. / Delete all flights that use aircrafts that were registered before 1975.
2. Airbike har foretaget en opgradering til flyene af typen A330, der øger hvert flys hastighed med 15% og antallet af pladser med 10%. Opdater alle fly af den type i overensstemmelse hermed. / Airbike has made an upgrade to the aircrafts of type A330 that increases each aircraft's speed by 15% and the number of seats by 10%. Update all aircrafts of that type accordingly.
3. Omdiriger! Alle flyvninger fra (`source`) København til (`destination`) London Stansted eller London Heathrow skal nu have destination mod (`destination`) London Luton i stedet. / Reroute! All flights from (`source`) Copenhagen to (`destination`) London Stansted or London Heathrow should now target (`destination`) London Luton instead.
4. Opret et materialiseret view for fly (med alle deres attributer), der flyver fra (`source`) eller til (`destination`) København. / Create a materialized view for aircrafts (with all their attributes) that fly from (`source`) or to (`destination`) Copenhagen.
5. COVID-19 angreb: Slet alle flyvninger! / COVID-19 strikes: Delete all flights!

Solution

1. ```
   DELETE FROM Flights
   WHERE aircraftNr IN (SELECT aircraftNr FROM Aircraft WHERE registeredYear < 1975)
   ```
2. ```
   UPDATE Aircraft
   SET speed = speed * 1.15 AND seats = seats * 1.1
   WHERE type = "A330"
   ```
3. ```
   UPDATE Flights
   SET destination = "London Luton"
   WHERE source = "Copenhagen" AND (destination = "London Stansted" OR "London Heathrow")
   ```
4. ```
   CREATE MATERIALIZED VIEW CPH_Aircraft AS
   SELECT A.*
   FROM Aircraft A NATURAL JOIN Flights F
   WHERE F.source = "Copenhagen" OR F.destination= "Copenhagen"
   ```
5. ```
   DELETE FROM Flights
   ```
   ```