

Danmarks  
Tekniske  
Universitet



---

# Database Project

---

*Gruppe - 14*

Introductory databases - 02327

*Forfattere*

Ali Hassan Khan - s224551

Mustafa Atici Keskin - s224955

Mattias Silas Bresdahl Ibsen - s235450

Hermon Abraham Asfaha - s224963

Davide Baudino - s224956

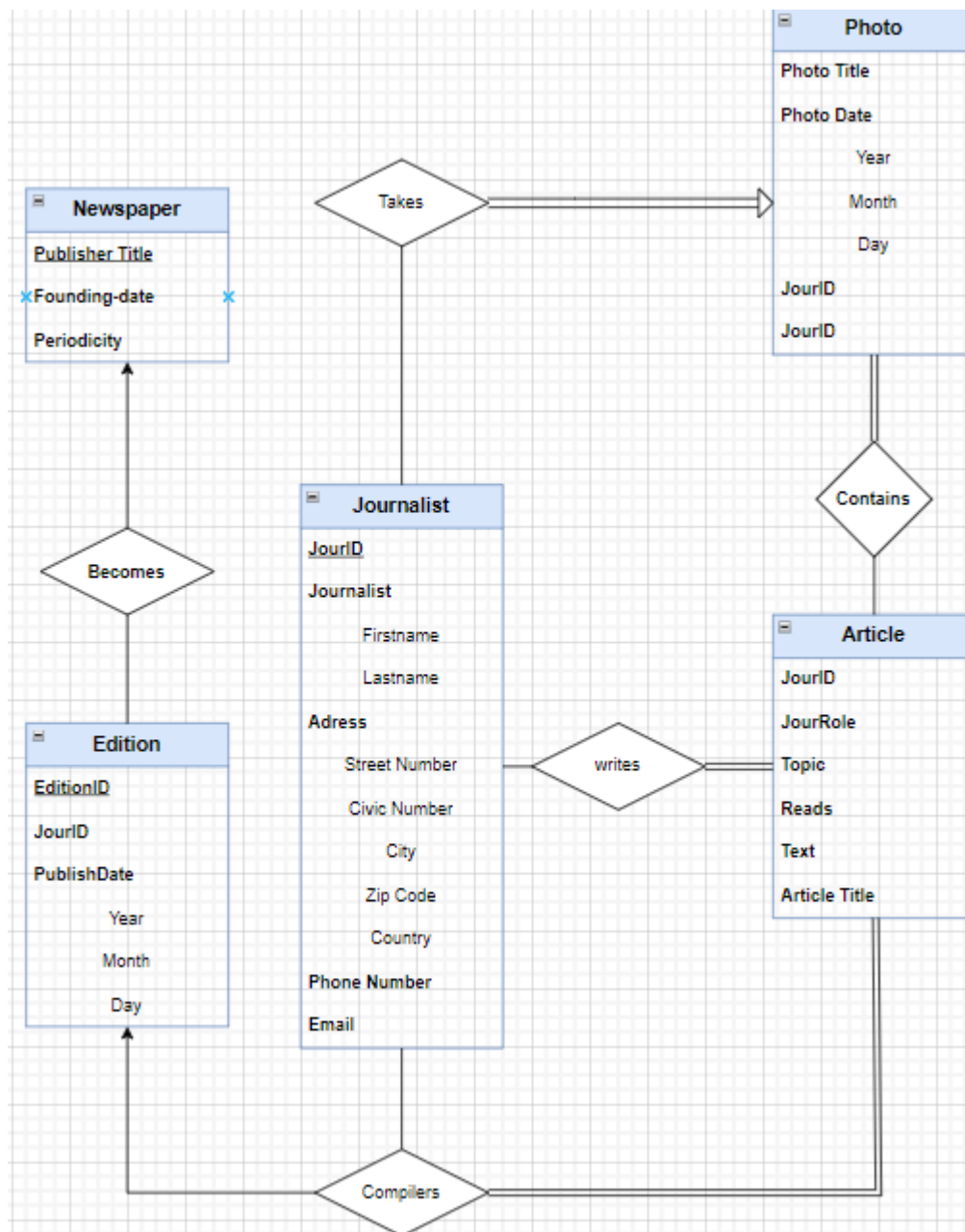
Dato: 05/04/2023

<b>Introduction.....</b>	<b>3</b>
<b>Conceptual Design.....</b>	<b>3</b>
<b>Logical design.....</b>	<b>4</b>
<b>Implementation.....</b>	<b>6</b>
<b>Database Instance.....</b>	<b>8</b>
<b>SQL table modifications.....</b>	<b>11</b>
<b>SQL data queries.....</b>	<b>12</b>
<b>SQL Programming.....</b>	<b>15</b>
<b>Java Database access.....</b>	<b>16</b>

## **Introduction**

The DKAvisen have asked for a database to keep track of the articles being published as well as the journalist who wrote it and the photos being shot. The DKAvisen have been facing a loss when it comes to their revenue so they want to keep track of the individual team performance of the workers as well as the amount of attraction there is to the different news articles they publish.

# Conceptual Design



## Entities & Attributes

The above picture showcases an uml diagram of our conceptual design over our database showing the different tables that are going to be created as well as the relationship between different tables. here is a list of the different tables and attributes

### Journalist:

This table contains the following attributes, {JourID, First Name, Last Name, Street Number, Civic Number, City, Zip code, Country, Phone Number, Email}

With the primary key being: JourID

Entity set: weak

**Photo:**

This table contains the following attributes, {PhotoID, Photo Date, Year, Month, Day, JourID, Photo Title}

With the primary key being: PhotoID

Entity set: weak

**Article:**

This table contains the following attributes, {Article Title, PhotoID, JourID, JourRole, Topic, Reads, Text}

With the primary key being: Article Title

Entity set: weak

**Edition:**

This table contains the following attributes, {EditionID, JourID, Publish Date, Year, Month, Day}

With the primary key being: EditionID

Entity set: weak

**Newspaper:**

This table contains the following attributes, {Publisher Title, Founding Date, Year, Month, Day}

With the primary key being: Publisher Title

Entity set: strong, because every key in this set is unique(not present anywhere else)

**Relationships:**

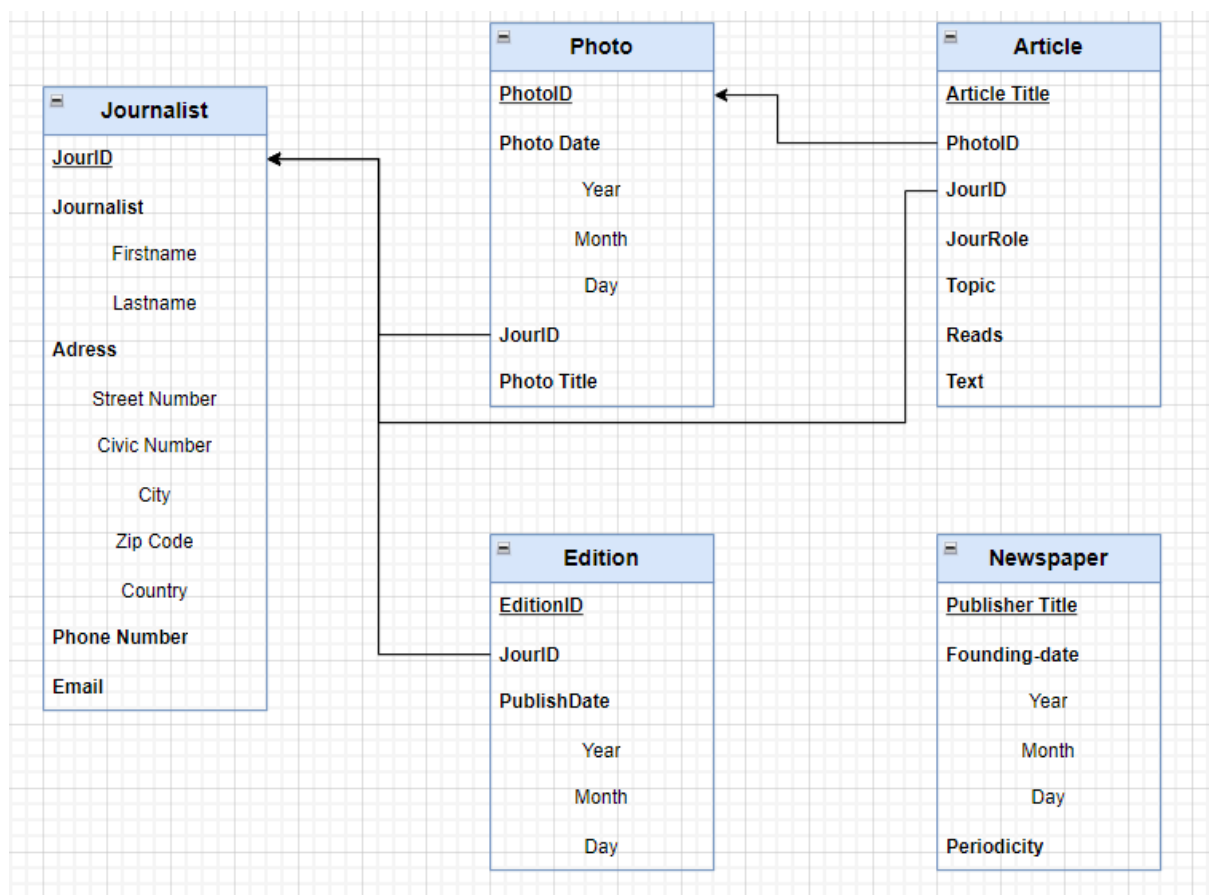
Becomes: the relationship between Newspaper and edition is shown through the becomes star which indicates a one to many relationship. This is because one newspaper can have many editions while each edition is only associated with exactly one newspaper. this relationship shows a partial participation from both entities

Writes: the relationship between the Journalist and Article entity is shown through the Writes star. This is a many to many relationship because one journalists can write many articles, and articles can have multiple writers, this would be partial participation from the journalist and total from article, because a journalist does not have to write many articles though every article needs at least one journalist

contains: this relationship between the Photo and Article entity is a many to many relationship with a total participation from photo while the Article entity being partial. This is shown because one article can contain multiple photos, but every article contains one photo.

Takes: is a many to one relationship because every photo requires a journalist, while every journalist does not have to take a photo showing total participation from the Photo entity and partial from the Journalist entity

Compilers: this relationship contains three entities. It is a many to one relationship between the entities Journalist and Article to the Edition entity. The participation of Journalist and Edition is partial while Article being total. This is because each edition can contain one or more news articles while one news article can only be present within one edition. It is also important to note that one edition can have multiple editors while each journalist can only write one editions.



The picture above is a logical diagram of our conceptual design with the focus being the relation between tables with their individual primary and foreign keys.

The entities shown in the diagram are connected, by, their different primary keys being foreign keys in other tables.

One example is the Edition table shown above, containing the foreign key JourID from the journalist showing the relationship between the two tables Edition and Journalist, because every edition has an Editor.

Another example is the relation between the tables Photo and Article as well as the tables Journalist and Article showcasing a table containing two foreign keys from different tables, one being the foreign key JourID and the other being PhotoID. The relation between Journalist and Article is established on the basis of every news article containing one or more writers(Journalist), and the relation between Photo and Article is established, on the basis that multiple news articles can contain the same photo

Another one is the relation between the tables Photo and Journalist with the table Photo containing the foreign key JourID. The relation between Journalist and Photo is established because a Photo is characterized by the Journalist who reported it.

At last, we have the Edition table which has established a relation to the Journalist table by using the JourID as a foreign key, because every Edition is characterized by the editor i.e. the journalist.

## Implementation

### Database

To start the implementation part by creating a database by using relation schemas from the previous part, we call the database Project. We create and name the database “projectGroup14” while at the same time using the database as shown below.

```
1 • create database projectGroup14;  
2 • use projectGroup14;
```

### Table

We will use the logical design to create the table by using the create table statement and then apply the different attributes in each table. The attributes have domains or fields that each describe a characteristic of the stored data. We have created 6 tables such as:

Journalist	Newspapers	Editions	Photo	Articles
------------	------------	----------	-------	----------

These tables are shown below and are each described.

## Journalist

This table is named Journalist and has 10 attributes. The primary key is JourID which contains a unique value. 7 of our attributes have the domain VARCHAR which is a variable character string with a fixed number of characters.

```

13 • CREATE TABLE Journalist
14 (
15     JourID          VARCHAR(100) NOT NULL,
16     firstname       VARCHAR(30) NOT NULL,
17     lastname        VARCHAR(100) NOT NULL,
18     StreetName      VARCHAR(250),
19     CivicNumber      INT,
20     City             VARCHAR(250),
21     ZipCode         INT,
22     Country         VARCHAR(250),
23     PhoneNumber     INT,
24     Email           VARCHAR(255),
25     primary key(JourID)
26 );

```

## Newspapers

This table is given the name Newspapers and contains 3 attributes. Publisher\_title is the primary key of this table. It contains a date domain in the FoundingDate attribute and two attributes with the same VARCHAR domain.

```

28 • CREATE TABLE newspapers
29 (
30     Publisher_title  VARCHAR(100),
31     FoundingDate     DATE,
32     Periodicity      VARCHAR(100),
33     primary key(Publisher_title)
34
35 );

```

## Editions



We have named this table Editions, and this has 3 attributes. The primary key of this table is EditionID. Other than the VARCHAR domain in the JourID attribute, Edition has the INT domain and PublishDate has the DATE domain. This also has a foreign key which is an attribute from another table. This foreign key is the attribute JourID from the Journalist table.

```
37 • CREATE TABLE Editions
38 (
39     JourID          VARCHAR(50),
40     EditionID       INT,
41     PublishDate     DATE,
42     PRIMARY KEY(EditionID),
43     FOREIGN KEY (JourID) REFERENCES Journalist(JourID)
44 )
45 ;
```

## Photo

This table is named Photo and has 4 attributes. Three attributes (PhotoID, title, JourID) have the VARCHAR domain and the primary key is PhotoID. This table has also a foreign key JourID which references the Journalist table.

```
47 • CREATE TABLE Photo
48 (
49     PhotoID         VARCHAR(100),
50     title           VARCHAR(100) NOT NULL,
51     PhotoDate       DATE,
52     JourID          VARCHAR(100) NOT NULL,
53     PRIMARY KEY(PhotoID),
54     FOREIGN KEY (JourID) REFERENCES Journalist(JourID)
55 )
```

## Articles

This table is named Articles and contains 7 attributes. The primary key is Title. This has two foreign keys one from references the journalist table and the attribute JourID and the other references from the Photo table and the attribute PhotoID. It also has an attribute called text which has a VARCHARA domain that has a high character count (10000).

```

57 • CREATE TABLE Articles
58 (
59     Title          VARCHAR(100),
60     JourID         VARCHAR(100) NOT NULL,
61     JourRole       VARCHAR(100) NOT NULL,
62     Topic          VARCHAR(100) NOT NULL,
63     ReadingCount   INT,
64     text           VARCHAR(10000),
65     PhotoID        VARCHAR(100) ,
66     PRIMARY KEY(Title),
67     FOREIGN KEY (JourID) REFERENCES Journalist(JourID),
68     FOREIGN KEY(PhotoID) REFERENCES Photo(PhotoID)
69 );

```

## Database Instance

The first image shows an “INSERT INTO” command adding records to the Journalist database table.

### Journalist

```

73 • INSERT INTO Journalist (JourID, FirstName, LastName, StreetName, CivicNumber, City, ZipCode, Country, PhoneNumber, Email)
74 VALUES
75 ('17-05-99-2049', 'Lars', 'Larsen', 'Magleparken', 45, 'Ballerup', 2750, 'Danmark', 35629185, 'Lars@larsen.dk'),
76 ('12-04-96-3147', 'Peter', 'Petersen', 'Lilleteoften', 127, 'Skovlunde', 2740, 'Danmark', 46730296, 'Peter@petersen.dk'),
77 ('20-10-95-4523', 'Emil', 'Emilsen', 'Blågårds Plads', 43, 'Nørrebro', 2200, 'Danmark', 57841307, 'Emil@emilsen.dk'),
78 ('3-11-94-2993', 'Anders', 'Andersen', 'Paltholmterrasserne', 200, 'Farum', 3520, 'Danmark', 68952418, 'Anders@andersen.dk'),
79 ('01-06-96-5041', 'Jens', 'Jensen', 'Tomsgårdsvej', 23, 'Nordvest', 2400, 'Danmark', 79053529, 'Jens@jensen.dk'),
80 ('02-07-97-3141', 'Mikkel', 'Nørgaard', 'Elverhøjen', 21, 'Herlev', 2730, 'Danmark', 80164630, 'Mikkel@nørgaard.dk'),
81 ('03-08-98-3040', 'Ida', 'Idasen', 'Tvedvangen', 37, 'Herlev', 2730, 'Danmark', 91275741, 'Ida@idasen.dk'),
82 ('24-05-77-2046', 'Pia', 'Kjærsgaard', 'Havnegade', 44, 'Strøby', 4671, 'Danmark', 2386852, 'Pia@kjærsgaard.dk'),
83 ('30-06-80-2248', 'Line', 'Madsen', 'Fynshovedvej', 24, 'Greve', 2670, 'Danmark', 13497963, 'Line@madsen.dk'),
84 ('12-10-88-4020', 'Stine', 'Madsen', 'Søndergade', 30, 'Brønderslev', 9700, 'Danmark', 24508074, 'Stine@madsen.dk'),
85 ('01-01-00-3733', 'Christoffer', 'Ankersen', 'Nørrebrogade', 56, 'Nørrebro', 2200, 'Danmark', 68127935, 'chris@ankersen.dk');

```

this query displays the outcome of a SELECT \* FROM Journalist query, reflecting the inserted data including personal and contact details of journalists.

```
select * from Journalist;
```

	JourID	firstname	lastname	StreetName	CivicNumber	City	ZipCode	Country	PhoneNumber	Email
▶	01-01-00-3733	Christoffer	Ankersen	Nørrebrogade	56	Nørrebro	2200	Danmark	68127935	chris@Ankersen.dk
	01-06-96-5041	Jens	Jensen	Tomsgårdsvej	23	Nordvest	2400	Danmark	79053529	Jens@jensen.dk
	02-07-97-3141	Mikkel	Nørgaard	Elverhøj	21	Herlev	2730	Danmark	80164630	Mikkel@nørgaard.dk
	03-08-98-3040	Ida	Idasen	Tvedvangen	37	Herlev	2730	Danmark	91275741	Ida@idasen.dk
	12-04-96-3147	Peter	Petersen	Lilletoften	127	Skovlunde	2740	Danmark	46730296	Peter@petersen.dk
	12-10-88-4020	Stine	Madsen	Søndergade	30	Brønderslev	9700	Danmark	24508074	Stine@madsen.dk
	17-05-99-2049	Lars	Larsen	Magleparken	45	Ballerup	2750	Danmark	35629185	Lars@larsen.dk
	20-10-95-4523	Emil	Emilsen	Blågårds Plads	43	Nørrebro	2200	Danmark	57841307	Emil@emilsen.dk
	24-05-77-2046	Pia	Kjaersgaard	Havnegade	44	Strøby	4671	Danmark	2386852	Pia@kjaersgaard.dk
	3-11-94-2993	Anders	Andersen	Paltholmterraserne	200	Farum	3520	Danmark	68952418	Anders@andersen.dk
	30-06-80-2248	Line	Madsen	Fynshovedvej	24	Greve	2670	Danmark	13497963	Line@madsen.dk

## Newspapers

The first image shows an “INSERT INTO ” command adding records to the newspapers database table.

```

87 • INSERT INTO newspapers (Publisher_title, FoundingDate, Periodicity)
88 VALUES
89 ('New York Times','18050103','Daily'),
90 ('The Global Ledger','18720504','Weekly'),
91 ('Daily Adventure','19061103','Evening Editions'),
92 ('The Sunday Echo','19500208','Weekly on Sunday'),
93 ('Morning Tribune','19950106','Daily');

```

This query displays the outcome of a SELECT \* FROM Newspapers query, reflecting the inserted data including Publisher title, FoundingDate, and Periodicity of the newspaper

```

• select * from newspapers;

```

	Publisher_title	FoundingDate	Periodicity
▶	Daily Adventure	1906-11-03	Evening Editions
	Morning Tribune	1995-01-06	Daily
	New York Times	1805-01-03	Daily
	The Global Ledger	1872-05-04	Weekly
	The Sunday Echo	1950-02-08	Weekly on Sunday

## EdItions

The image shows an “INSERT INTO” command adding records to the Editions database table.

```

95 • INSERT INTO Editions(JourID,EditionID,PublishDate)
96 VALUES
97 ('17-05-99-2049',5003,20240714),
98 ('12-04-96-3147',6579,20230925),
99 ('20-10-95-4523',4352,20241012),
100 ('3-11-94-2993',2300,20240423),
101 ('01-06-96-5041',2770,20240511),
102 ('02-07-97-3141',3400,20240612),
103 ('03-08-98-3040',1200,20241211),
104 ('24-05-77-2046',3200,20240115),
105 ('30-06-80-2248',8700,20240228),
106 ('12-10-88-4020',9200,20240514),
107 ('01-01-00-3733',3700,20240401),
108 ('01-06-96-5041',9400,20240104);

```

This query displays the outcome of a `SELECT * FROM Editions` query, reflecting the inserted data including JourId and date of the edition.

```
select * from Editions;
```

	JourID	EditionID	PublishDate
►	03-08-98-3040	1200	2024-12-11
	3-11-94-2993	2300	2024-04-23
	01-06-96-5041	2770	2024-05-11
	24-05-77-2046	3200	2024-01-15
	02-07-97-3141	3400	2024-06-12
	01-01-00-3733	3700	2024-04-01
	20-10-95-4523	4352	2024-10-12
	17-05-99-2049	5003	2024-07-14
	12-04-96-3147	6579	2023-09-25
	30-06-80-2248	8700	2024-02-28
	12-10-88-4020	9200	2024-05-14
	01-06-96-5041	9400	2024-01-04

## SQL table modifications

### Insert

This Command allows us to insert a new row in the query and add the value for the new row

```

164 • INSERT INTO Articles (Title, JourID, JourRole, Topic, ReadingCount, Text, PhotoID)
165 VALUES ('New Article Title', '17-05-99-2049', 'Author', 'Technology', 0, 'This is the text of the new article.', NULL);
166 • select * from articles;

```

	Title	JourID	JourRole	Topic	ReadingCount	text	PhotoID
▶	A new Planet	24-05-77-2046	Author	Space	1000	NASA have a discovered 10th planet in our sola...	NULL
	arctic melting	17-05-99-2049	Author	Global news	1	The arctic ice has lost a third of its mass over th...	2021-06-05-antarctica-melting
	elder people 's loneliness is at its peak	01-06-96-5041	Editor	community	500	Down at the nursing home in the republic, chan...	2023-05-12-Singles-in-nursing-home
	Katt Williams exposed hollywood	01-01-00-3733	Editor	entertainment	350	famous comedian Katt Williams exposed critical a...	NULL
	kids spent longer time in front of screen than sl...	3-11-94-2993	Author	upbringing	5	In a recent research done by the university of ...	2022-07-11-Screen-eyes
	Madrid Victory	30-06-80-2248	Editor	Sports	200	Real Madrid won against rivals Barcelona in dos...	NULL
	Most played games in the year 2023	24-05-77-2046	Editor	Video Games	420	Here is a list showcasing the top most played ga...	NULL
	New Article Title	17-05-99-2049	Author	Technology	0	This is the text of the new article.	NULL
	New Plants	12-10-88-4020	Author	Nature	1500	DTU	NULL
	Remote Work Exposes Digital Divide	01-06-96-5041	Editor	Society	375	The widespread shift to remote work during the...	NULL
	Sorrow over the loss of the queen	12-04-96-3147	Editor	Royals	0	The people of The United Kingdom is enjoined in...	2020-12-02-The-queen-gets-carried
	student rally up in protest	20-10-95-4523	Leader	education	6	students tired of overwork, getting 0 hour slee...	2022-11-24-With-banners-and-flags
	Tesla release new car	03-08-98-3040	Author	Technology	800	Tesla has released a new car. This new car no l...	NULL
	University has new student-prgoram	02-07-97-3141	Editor	Education	400	The University has a new program designed for ...	NULL
	World Cup 2026	01-01-00-3733	leader	Sports	1065676	The whole globe is in shock, by the establishme...	NULL

## Delete

This command is used to delete an articles from the query articles with the name “ arctic melting”

```
168 • DELETE FROM Articles where Title = 'arctic melting';
169 • select * from articles;
```

	Title	JourID	JourRole	Topic	ReadingCount	text	PhotoID
▶	A new Planet	24-05-77-2046	Author	Space	1000	NASA have a discovered 10th planet in our sola...	NULL
	elder people 's loneliness is at its peak	01-06-96-5041	Editor	community	500	Down at the nursing home in the republic, chan...	2023-05-12-Singles-in-nursing-home
	Katt Williams exposed hollywood	01-01-00-3733	Editor	entertainment	350	famous comedian Katt Williams exposed critical a...	NULL
	kids spent longer time in front of screen than sl...	3-11-94-2993	Author	upbringing	5	In a recent research done by the university of ...	2022-07-11-Screen-eyes
	Madrid Victory	30-06-80-2248	Editor	Sports	200	Real Madrid won against rivals Barcelona in dos...	NULL
	Most played games in the year 2023	24-05-77-2046	Editor	Video Games	420	Here is a list showcasing the top most played ga...	NULL
	New Article Title	17-05-99-2049	Author	Technology	0	This is the text of the new article.	NULL
	New Plants	12-10-88-4020	Author	Nature	1500	DTU	NULL
	Remote Work Exposes Digital Divide	01-06-96-5041	Editor	Society	375	The widespread shift to remote work during the...	NULL
	Sorrow over the loss of the queen	12-04-96-3147	Editor	Royals	0	The people of The United Kingdom is enjoined in...	2020-12-02-The-queen-gets-carried
	student rally up in protest	20-10-95-4523	Leader	education	6	students tired of overwork, getting 0 hour slee...	2022-11-24-With-banners-and-flags
	Tesla release new car	03-08-98-3040	Author	Technology	800	Tesla has released a new car. This new car no l...	NULL
	University has new student-prgoram	02-07-97-3141	Editor	Education	400	The University has a new program designed for ...	NULL
	World Cup 2026	01-01-00-3733	leader	Sports	1065676	The whole globe is in shock, by the establishme...	NULL

## Update

The Command used in the image updates the query Articles and sets the reading count to 100 for the article with the title “ Sorrow over the loss of the queen”

Then all columns for the query **Article** is selected.

```
171 • UPDATE Articles SET ReadingCount = 100 WHERE Title = 'Sorrow over the loss of the queen';
172 • select * from articles;
```

	Title	JourID	JourRole	Topic	ReadingCount	text	PhotoID
▶	A new Planet	24-05-77-2046	Author	Space	1000	NASA have a discovered 10th planet in our sola...	NULL
	elder people 's loneliness is at its peak	01-06-96-5041	Editor	community	500	Down at the nursing home in the republic, chan...	2023-05-12-Singles-in-nursing-home
	Katt Williams exposed hollywood	01-01-00-3733	Editor	entertainment	350	famous comedian Katt Williams exposed critical a...	NULL
	kids spent longer time in front of screen than sl...	3-11-94-2993	Author	upbringing	5	In a recent research done by the university of ...	2022-07-11-Screen-eyes
	Madrid Victory	30-06-80-2248	Editor	Sports	200	Real Madrid won against rivals Barcelona in dos...	NULL
	Most played games in the year 2023	24-05-77-2046	Editor	Video Games	420	Here is a list showcasing the top most played ga...	NULL
	New Article Title	17-05-99-2049	Author	Technology	0	This is the text of the new article.	NULL
	New Plants	12-10-88-4020	Author	Nature	1500	DTU	NULL
	Remote Work Exposes Digital Divide	01-06-96-5041	Editor	Society	375	The widespread shift to remote work during the...	NULL
	Sorrow over the loss of the queen	12-04-96-3147	Editor	Royals	100	The people of The United Kingdom is enjoined in...	2020-12-02-The-queen-gets-carried
	student rally up in protest	20-10-95-4523	Leader	education	6	students tired of overwork, getting 0 hour slee...	2022-11-24-With-banners-and-flags
	Tesla release new car	03-08-98-3040	Author	Technology	800	Tesla has released a new car. This new car no l...	NULL
	University has new student-prgoram	02-07-97-3141	Editor	Education	400	The University has a new program designed for ...	NULL
	World Cup 2026	01-01-00-3733	leader	Sports	1065676	The whole globe is in shock, by the establishme...	NULL

## SQL data queries

For each topic, show the most read news article:

This query aims to find the most read news article for each topic category. It groups articles by topic and retrieves the one with the highest number of reads for each topic.

```
SELECT Topic, Title, MAX(ReadingCount) AS MaxReads
FROM Articles
GROUP BY Topic;
```

	Topic	Title	MaxReads
▶	community	elder people 's loneliness is at its peak	500
	education	student rally up in protest	400
	Global news	arctic melting	1
	Nature	New Plants	1500
	Royals	Sorrow over the loss of the queen	0

Show the top 10 journalists whose articles, overall, attracted the highest number of reads:

This query identifies the top 10 journalists whose articles collectively garnered the highest number of reads. It sums up the total reads for each journalist's articles and displays the top 10 based on this metric.

```
SELECT JourID, SUM(ReadingCount) AS TotalReads
FROM Articles
GROUP BY JourID
ORDER BY TotalReads DESC
LIMIT 10;
```

	JourID	TotalReads
►	12-10-88-4020	1500
	24-05-77-2046	1000
	03-08-98-3040	800
	01-06-96-5041	500
	02-07-97-3141	400

**Show reporters whose photos were never used more than once:**

Here, we're interested in finding reporters whose photos were used only once, indicating that their photos were not reused across multiple news articles.

```
SELECT JourID
FROM Photo
GROUP BY JourID
HAVING COUNT(PhotoID) = 1;
```

	JourID
►	01-06-96-5041
	12-04-96-3147
	17-05-99-2049
	20-10-95-4523
	3-11-94-2993

**Identify which topics, overall, attracted fewer reads than the average:**

This query calculates the average number of reads for each topic and compares it to the overall average across all topics. It identifies topics with average reads lower than the overall average. Identify which journalists were both writers and reporters, having shot at least a photo that

```
SELECT Topic, AVG(ReadingCount) AS AvgReads
FROM Articles
GROUP BY Topic
HAVING AVG(ReadingCount) < (SELECT AVG(ReadingCount) FROM Articles);
```

Topic	AvgReads
education	203.0000
Global news	1.0000
Royals	0.0000
Sports	200.0000
upbringing	5.0000

**Was used for a news article they wrote:**



The objective here is to find journalists who served both as writers (authors) and reporters. Specifically, we're looking for those who not only wrote articles but also contributed photos used in the articles they authored.

```
SELECT DISTINCT a.JourID
FROM Articles a
INNER JOIN Photo p ON a.JourID = p.JourID
WHERE a.JourRole = 'Author' AND p.JourID = a.JourID;
```

JourID
17-05-99-2049
3-11-94-2993

## SQL Programming

**Functions in SQL:** In relational database management systems, functions are defined as deterministic operations that accept parameters, execute logic, and return a value. The function *CalculateAverageReads* is designed to compute the mean read count for articles within a specified subject domain. When invoked with a parameter, such as *CalculateAverageReads('Technology')*, the function returns the average read count for articles categorized under 'Technology'.

```

DROP FUNCTION IF EXISTS CalculateAverageReadingCount;
DELIMITER $$
CREATE FUNCTION CalculateAvgReadsForTopic(topic VARCHAR(255)) RETURNS INT
BEGIN
    DECLARE avg_ReadingCount INT;
    SELECT AVG(ReadingCount) INTO avg_Readingcount FROM Article WHERE Topic = topic;
    RETURN avg_ReadingCount;
END;
DELIMITER $$

```

**Procedures in SQL:** Stored procedures represent a set of precompiled SQL statements that encapsulate complex database operations. The *GetArticlesByJournalist* procedure retrieves articles authored by a journalist identified by their unique ID. Executing CALL *GetArticlesByJournalist*('17-05-99-2049') would yield a result set of articles penned by the journalist with the ID '17-05-99-2049'.

```

CREATE PROCEDURE GetArticlesByJournalist(IN journalist_id VARCHAR(20))
BEGIN
    SELECT * FROM Article WHERE JourID = journalist_id;
END;

```

**Triggers in SQL:** Triggers are automated mechanisms triggered by specific database events. For instance, the *AfterInsertArticle* trigger is designed to respond to insertions into the Article table. Upon the insertion of a new article, the trigger performs an action, such as logging the event to an *AuditTrail* table. This trigger ensures that every insertion into the Article table is accompanied by a corresponding audit record, thereby maintaining data integrity and providing a transaction history.

```

CREATE TRIGGER AfterInsertArticle
AFTER INSERT ON Article
FOR EACH ROW
BEGIN
    INSERT INTO AuditTrail (Action, TableName, ArticleID) VALUES ('INSERT', 'Article', NEW.ArticleID);
END;

```

## Java Database access

Before we explain the Java code, we want to make it clear that we will not be diving into handed out code, which we haven't changed.

We first set up a connection to the database via a local host.

Then we set up sql statements for inserting Journalist and Photo.

```
// Open a connection to the database
Connection conn = DriverManager.getConnection(CONNECTION_STRING, USERNAME, PASSWORD);

// Prepare the SQL statement
String sql = "INSERT IGNORE INTO journalist (JourID, firstname, lastname, StreetName, CivicNumber, ZipCode, city) VALUES (?, ?, ?, ?, ?, ?, ?)";
PreparedStatement statement = conn.prepareStatement(sql);
String insertPhotoSql = "INSERT IGNORE INTO photo (photoid, title, photodate, JourID) VALUES (?, ?, ?, ?)";
PreparedStatement PhotoStmt = conn.prepareStatement(insertPhotoSql);

// Disable auto-commit for batch execution
conn.setAutoCommit(autoCommit:false);
```

Here use the functions in the Reporter class and the Photo class to get different values.

```
try {
    // Loop through the list and add batches
    for (PhotoAndReporter photoAndReporter : photosAndReporters) {
        Reporter reporter = photoAndReporter.getReporter();
        Photo photos = photoAndReporter.getPhoto();
        statement.setInt(parameterIndex:1, reporter.getCPR());
        statement.setString(parameterIndex:2, reporter.getFirstName());
        statement.setString(parameterIndex:3, reporter.getLastName());
        statement.setString(parameterIndex:4, reporter.getStreetName());
        statement.setInt(parameterIndex:5, reporter.getCivicNumber());
        statement.setInt(parameterIndex:6, reporter.getZIPCode());
        statement.setString(parameterIndex:7, reporter.getCity());
        PhotoStmt.setString(parameterIndex:1, photos.getPhotoID());
        PhotoStmt.setString(parameterIndex:2, photos.getTitle());
        PhotoStmt.setString(parameterIndex:3, photos.getDate());
        PhotoStmt.setInt(parameterIndex:4, reporter.getCPR());
        // Add to batch
        statement.addBatch();
        PhotoStmt.addBatch();
    }

    // Execute batch
    statement.executeBatch();
    PhotoStmt.executeBatch();
    // Commit changes
    conn.commit();
}
```

One of the challenges was retrieving the PhotoID data since it was not present in the CSV file. So we concatenated the title of the photo with the date.

There were some obstacles in the way.

```
private final String title;
private final Date date;
private final String PhotoID;

SimpleDateFormat dateFormat = new SimpleDateFormat(pattern:"yyyy-MM-dd");
public Photo(String title, Date date, String PhotoID) {
    this.title = title;
    this.date = date;
    this.PhotoID = PhotoID;
}

public String getTitle() {
    return title;
}
public String getDate() {
    return dateFormat.format(date);
}

PhotoAndReporter far = new PhotoAndReporter(title, date, cpr, firstName, lastName, streetName, civicNumber, zipCode, city,PhotoID);
```

We created a new string variable PhotoID, and formatted the date. The format returns the date as a string. We then extended the Photo class to have another function that would give us the PhotoID and that also concatenates the ID for us. We also change in line 62 in the PhotosAndReportersLoader file the instance of PhotoAndReporter class

```
public String getPhotoID(){
    return dateFormat.format(date) + '-' + title ;
}
```

Besides these changes, we haven't changed anything in the received code.

These are the tables before we use java to insert the new instances

JourID	firstname	lastname	StreetName	CivicNumber	City	ZipCode	Country	PhoneNumber	Email
01-01-00-3733	Christoffer	Ankersen	Nørrebrogade	56	Nørrebro	2200	Danmark	68127935	chris@Ankersen.dk
01-06-96-5041	Jens	Jensen	Tomsgårdsvej	23	Nordvest	2400	Danmark	79053529	Jens@jensen.dk
02-07-97-3141	Mikkel	Nørgaard	Elverhøjen	21	Herlev	2730	Danmark	80164630	Mikkel@nørgaard.dk
03-08-98-3040	Ida	Idasen	Tvedvangen	37	Herlev	2730	Danmark	91275741	Ida@idasen.dk
12-04-96-3147	Peter	Petersen	Lilletoften	127	Skovlunde	2740	Danmark	46730296	Peter@petersen.dk
12-10-88-4020	Stine	Madsen	Søndergade	30	Brønderslev	9700	Danmark	24508074	Stine@madsen.dk
17-05-99-2049	Lars	Larsen	Magleparken	45	Ballerup	2750	Danmark	35629185	Lars@larsen.dk
20-10-95-4523	Emil	Emilsen	Blågårds Plads	43	Nørrebro	2200	Danmark	57841307	Emil@emilsen.dk
24-05-77-2046	Pia	Kjaersgaard	Havnegade	44	Strøby	4671	Danmark	2386852	Pia@kjaersgaard.dk
3-11-94-2993	Anders	Andersen	Paltholmterrasserne	200	Farum	3520	Danmark	68952418	Anders@andersen.dk
30-06-80-2248	Line	Madsen	Fynshovedvej	24	Greve	2670	Danmark	13497963	Line@madsen.dk

	PhotoID	title	PhotoDate	JourID
▶	2020-12-02-The-queen-gets-carried	The queen gets carried	2020-12-02	12-04-96-3147
	2021-06-05-antarctica-melting	antarctica melting	2021-06-05	17-05-99-2049
	2022-07-11-Screen-eyes	Screen eyes	2022-07-11	3-11-94-2993
	2022-11-24-With-banners-and-flags	With banners and flags	2022-11-24	20-10-95-4523
	2023-05-12-Singles-in-nursing-home	Singles in nursing home	2023-05-12	01-06-96-5041

This is the tables after we load the instances from the CSV file

We didn't format the CPR numbers from the csv file since they aren't formatted like the usual Danish CPR numbers.

JourID	firstname	lastname	StreetName	CivicNumber	City	ZipCode	Country	PhoneNumber	Email
▶ 01-01-00-3733	Christoffer	Ankersen	Nørrebrogade	56	Nørrebro	2200	Danmark	68127935	chris@Ankersen.dk
01-06-96-5041	Jens	Jensen	Tomsgårdsvej	23	Nordvest	2400	Danmark	79053529	Jens@jensen.dk
02-07-97-3141	Mikkel	Nørgaard	Elverhøjen	21	Herlev	2730	Danmark	80164630	Mikkel@nørgaard.dk
03-08-98-3040	Ida	Idasen	Tvedvangen	37	Herlev	2730	Danmark	91275741	Ida@idasen.dk
10103040	Mark	Miller	Nulvej	132	Kgs. Lyngby	2800	NULL	NULL	NULL
10203344	Lukas	Laas	Åstergaardsvej	12	Gentofte	2820	NULL	NULL	NULL
10204410	Nick	Nassar	Vestergaardsvej	13	Virum	2830	NULL	NULL	NULL
12-04-96-3147	Peter	Petersen	Lilletoften	127	Skovlunde	2740	Danmark	46730296	Peter@petersen.dk
12-10-88-4020	Stine	Madsen	Søndergade	30	Brønderslev	9700	Danmark	24508074	Stine@madsen.dk
17-05-99-2049	Lars	Larsen	Magleparken	45	Ballerup	2750	Danmark	35629185	Lars@larsen.dk
20-10-95-4523	Emil	Emilsen	Blågårds Plads	43	Nørrebro	2200	Danmark	57841307	Emil@emilsen.dk
20208981	Pia	Pabst	Hovedgade	9	Nårum	2850	NULL	NULL	NULL
24-05-77-2046	Pia	Kjaersga...	Havnegade	44	Strøby	4671	Danmark	2386852	Pia@kjaersgaard.dk
3-11-94-2993	Anders	Andersen	Palholmterrass...	200	Farum	3520	Danmark	68952418	Anders@andersen.dk
30-06-80-2248	Line	Madsen	Fynshovedvej	24	Greve	2670	Danmark	13497963	Line@madsen.dk
30302125	Olga	Owens	Nybrovej	28	Holte	2840	NULL	NULL	NULL
★ NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	PhotoID	title	PhotoDate	JourID
▶	2004-05-14-Royal Wedding	Royal Wedding	2004-05-14	10103040
	2020-03-11-COVID-19 Restrictions	COVID-19 Restrictions	2020-03-11	30302125
	2020-12-02-The-queen-gets-carried	The queen gets carried	2020-12-02	12-04-96-3147
	2021-06-05-antarctica-melting	antarctica melting	2021-06-05	17-05-99-2049
	2021-10-09-Fatal Car Accident	Fatal Car Accident	2021-10-09	30302125
	2021-11-10-Pottery Exhibit	Pottery Exhibit	2021-11-10	10203344
	2022-05-29-Superliga Final Match	Superliga Final Match	2022-05-29	20208981
	2022-07-11-Screen-eyes	Screen eyes	2022-07-11	3-11-94-2993
	2022-07-18-SAS Strike	SAS Strike	2022-07-18	30302125
	2022-11-01-Election Day	Election Day	2022-11-01	10204410
	2022-11-24-With-banners-and-flags	With banners and flags	2022-11-24	20-10-95-4523
	2023-03-06-Snowstorm	Snowstorm	2023-03-06	10103040
	2023-05-12-Singles-in-nursing-home	Singles in nursing home	2023-05-12	01-06-96-5041
★	NULL	NULL	NULL	NULL