Name:

Instructions

• Try to answer all the questions using what you have learned in class. Keep hard questions until the end.

• When writing a query, write the query in a way that it would work over all possible database instances and not just for the given example instance!

• The exam is closed book and one page of note is allowed

• For relational algebra questions assume set semantics! Consider the following database schema and example instance for a car database

An artist

ArtistId	Name
1	AC/DC
2	Accept
3	Aerosmith
4	Alanis Morissette
5	Alice In Chains
6	Antônio Carlos Jobim
7	Apocalyptica
8	Audioslave

Album

AlbumId	Title	ArtistId
1	For Those About To Rock We Salute You	1
2	Balls to the Wall	2
3	Restless and Wild	2
4	Let There Be Rock	1
5	Big Ones	3
6	Jagged Little Pill	4
7	Facelift	5
8	Warner 25 Anos	6

MediaType

MediaTypeId	Name
1	MPEG audio file
2	Protected AAC audio file
3	Protected MPEG-4 video file
4	Purchased AAC audio file
5	AAC audio file

Track

TrackId	Name	AlbumId	MediaTypeId
1	For Those About To Rock (We Salute You)	1	1
2	Balls to the Wall	2	2
3	Fast As a Shark	3	2
4	Restless and Wild	3	2
5	Princess of the Dawn	3	2
6	Put The Finger On You	1	1
7	Let's Get It Up	1	1
8	Inject The Venom	1	1
9	Snowballed	1	1
10	Evil Walks	1	1

Invoice

	InvoiceLineId	InvoiceId	TrackId	UnitPrice	Quantity
1	1	1	2	0.9900	1
2	2	1	4	0.9900	1
3	3	2	6	0.9900	1
4	4	2	8	0.9900	1
5	5	2	10	0.9900	1
6	6	2	12	0.9900	1
7	7	3	16	0.9900	1
8	8	3	20	0.9900	1

ArtistID is a primary key for Artist and a foreign key of (referenced) relation Album
MediaTypeId is a primary key for MediaType and a foreign key of relation Track
AlbumId is a primary key for Album and a foreign key of relation
TrackId is a primary key for track
All foreign keys have been created with the CASCADE option.

Part 1 Relational Algebra (Total: 28 Points)
Question 1.1 (7 points) Write a relational algebra that return all media type MPEG audio or MPEG 4 video

$$\sigma_{Name='MPEGaudifile'VName='\Pr otectedvMPEG-4videofle}(MediaType)$$

Question 1.2 (10 points) 1.2 Write a relational algebra that return the track with the associated artist AC/DC and Accept

$$\pi_{Artist.Name,Track.Name}\sigma_{Artist.Name='AC/DC'VArftist.Name='Accept'}(Artist\infty Album\infty Track)$$

Question 1.3 (11 points) 1.3 Write a relational algebra that return most expensive track in each album

$$\pi_{AlbumId,Track.Name\ AlbumId}G_{\max(unit\ \Pr ice)}(Invoice\infty Track)$$

Part 2 SQL DDL
Question 2.1 (16 points)
Write an SQL DDL statement that creates a new relation sponsorship that records information about an artist sponsor . This relation should have attributes sponsorName, contractNumber, contractLength, artistID, artistName . A sponsorship relation is uniquely identified by the combination of sponsorName and contractNumber. Attribute artistID is a foreign key to relation Artist. When an Artist that is currently assigned to sponsorship is deleted from the database, then the artistID attribute of sponsorship should be set to NULL. If an artistName of an Artist is updated, then the artistName attribute of all sponsorship the artistName is assigned to should be updated too. Note that contractLength cannot be negative and cannot be NULL. Contract length has to be positive.

```
CREATE TABLE sponsership (
sponsorName VARCHAR ( 30 ) ,
contractNumber INT ,
contractLength INT not null,
artistID INT ,
artistName VARCHAR ( 1 2 ) ON UPDATE CASCADe
PRIMARY KEY ( sponsorName, contractNumber ) ,
 FOREIGN KEY ( artistID ) REFERENCES artist ON DELETE SET NULL  ,
 CHECK ( contractLength >= 0. )
```

Question 3.1 (10 points) Write an SQL query that returns the artistId and albumid for the “Big Ones” album

```
select Album.AlbumId,artistId
from Artist join Album on Artist.ArtistId=Album.ArtistId;
where Album.Title='Big Ones'
```

Question 3.2 (10 points) Write an SQL query that returns total price for each album

```
select AlbumId,sum(InvoiceLine.UnitPrice)
from track  join InvoiceLine on track.TrackId=InvoiceLine.TrackId
group by AlbumId
```

Question 3.3 (15 points) Write an SQL query that returns the track whose MediaType is not MPEG4

```
select trackId, Track.Name, MediaType.Name
from Track join MediaType on Track.MediaTypeId =MediaType.MediaTypeId
where MediaType.Name!='Protected MPEG-4 video file' AND MediaType.Name!= 'MPEG audio
file'
```

Alternative

```
select trackId, Track.Name, MediaType.Name
from Track join MediaType on Track.MediaTypeId =MediaType.MediaTypeId
where MediaType.Name not like '%MPEG%' ;
```

Question 4.1 (13 Points)

Write an SQL statement that deletes all the album that don't have a track .

```
Delete from album
Where AlbumId not in (select AlbumId
                      from Track)
```

Question 4.2 Write an SQL statement that increase all unitprice by 2 (7 points)

```
upDATE InvoiceLine set UnitPrice=(SELECT UnitPrice from Invoice)+2 ;
```