

1. Relational Algebra and SQL

1.1 a)

```
1  -- 1.
2  SELECT DISTINCT b.title
3  FROM books AS b
4  JOIN authors AS a ON b.author = a.aid
5  WHERE b.year > 1989 AND a.salary != 1500;

1  -- 2.
2  SELECT b.genre
3  FROM (
4      SELECT *
5      FROM borrow b
6      JOIN books ON b.book = books.bid
7      WHERE b.borrow_date = '16.05.2024'
8      GROUP BY b.genre
9      HAVING COUNT(*) > 10
10 )
```

1.2 b)

1. $\pi * (\sigma_{late_fees=0}members)(\sigma_{city \neq 'Saarbruecken'}libraries) \bowtie (\sigma_{member=mid}membership)$
2. $\pi_{avg_late_fees}()$

2. Natural Language and SQL

2.1 a)

```
1  -- 1.
2  SELECT DISTINCT b.car
3  FROM buy b
4  LEFT JOIN repair r ON b.car = r.car AND r.reason = 'broken clutch'
5  WHERE YEAR(b.date_of_purchase) < 2015 AND r.car IS NULL;

1  -- 2.
2  SELECT m.mid
3  FROM mechanics m
4  JOIN repair r ON m.mid = r.mechanic
5  JOIN cars c ON r.car = c.id
6  GROUP BY m.mid
7  HAVING COUNT(r.start) > 5 AND MAX(DATEDIFF(MONTH, r.start, r.end)) < 6 AND c.
   horsepower = 250;
8
```

2.2 b)

1. Find the unique cities and salaries of mechanics who have serviced more than one Volkswagen Golf GTI vehicle
2. Give a list of full names of the top ten earning mechanics, sorted by decreasing age, who earn more than 4000 per year and don't have the last name Smith.

3. SQL Debugging

3.1

- GROUP BY should be after the WHERE
- JOIN should use 'c.id' and 'p.cameraId'
- NULL AS NOT_NULL doesn't provide any meaningful information for this query
- 'location' is an ambiguous column definition. Should be 'p.location'.

```
1  -- Corrected Query
2  SELECT p.location
3  FROM cameras AS c
4  JOIN photos AS p ON c.id = p.cameraId
5  WHERE location LIKE '%bruecken'
6  GROUP BY location;
```

3.2

- 'measureOfAge' is an alias for an aggregated column, so it can't be used in the GROUP BY clause.
- JOIN on employees is needed for 'experience' in the WHERE clause.
- sum() on 's.numGreyHairs' is unnecessary, since it will only sum up single values.

```
1  -- Corrected Query
2  SELECT s.numGreyHairs AS measureOfAge
3  FROM seniors AS s
4  JOIN employees AS e ON s.employeeId = e.personId
5  WHERE e.experience > 42;
```

3.3

- HAVING is used on GROUP BY aggregations, should be WHERE instead.
- Without aggregation sum() on 'experience' is unnecessary, since it will only sum up single values.

```
1  -- Corrected Query
2  SELECT salary
3  FROM employees
4  WHERE experience > 5;
```