Leo Forster
Maximilian Prinz
Bastian Simon

1. Relational Algebra and SQL

1.1 a)

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

```sql
-- 2.
SELECT b.genre
FROM (
  SELECT *
  FROM borrow b
  JOIN books ON b.book = books.bid
  WHERE b.borrow_date = '16.05.2024'
  GROUP BY b.genre
  HAVING COUNT(*) > 10
)
```

1.2 b)

1. $\sigma_{city \neq 'Saarbruecken'}(libraries \bowtie_{lid=libraries}(membership \bowtie_{member=mid} members))$
2. $R_1 := members \bowtie_{favorite\_author=aid}(\sigma_{salary>2000} authors)$
   $R_2 := \sigma_{sum(late\_fees)=100}(\gamma_{reservation\_date,sum(late\_fees)}(reserve \bowtie_{member=mid} R_1))$
   $\gamma_{avg(late\_fees)} R_2$

2. Natural Language and SQL

2.1 a)

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

```sql
-- 2.
SELECT m.mid
FROM mechanics m
JOIN repair r ON m.mid = r.mechanic
JOIN cars c ON r.car = c.id
GROUP BY m.mid
```

Leo Forster
Maximilian Prinz
Bastian Simon

```sql
HAVING COUNT(r.start) > 5 AND MAX(DATEDIFF(MONTH, r.start, r.end)) < 6 AND c.
  horsepower = 250;

```

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 ambigouus colum definition. Should be 'p.location'.

```sql
-- Corrected Query
SELECT p.location
FROM cameras AS c
JOIN photos AS p ON c.id = p.cameraId
WHERE location LIKE '%bruecken'
GROUP BY location;
```

3.2

- 'measureOfAge' is an alias for a 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.

Leo Forster
Maximilian Prinz
Bastian Simon

```sql
-- Corrected Query
SELECT s.numGreyHairs AS measureOfAge
FROM seniors AS s
JOIN employees AS e ON s.employeeId = e.personId
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.

```sql
-- Corrected Query
SELECT salary
FROM employees
WHERE experience > 5;
```