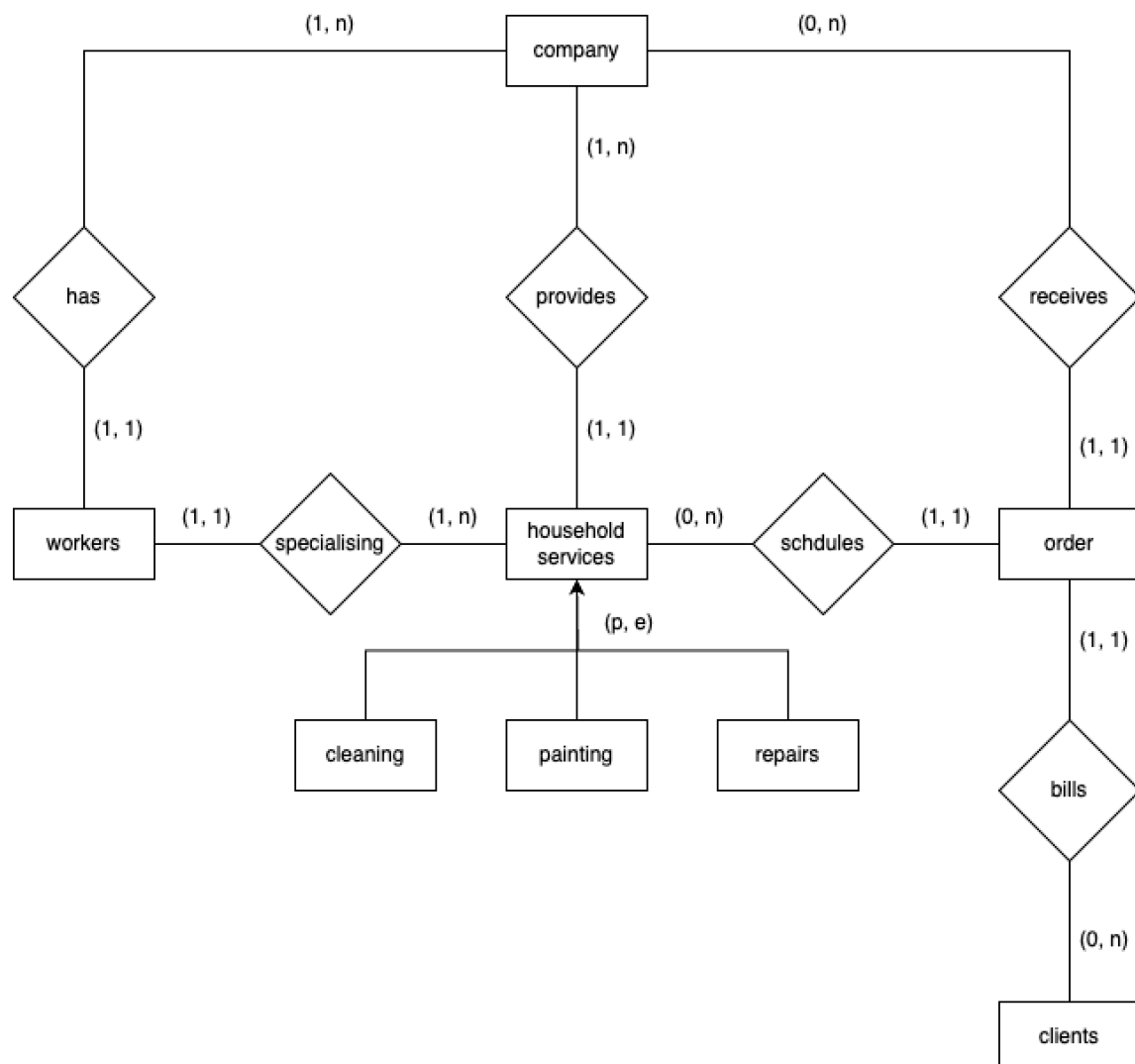


ID 2546379

# Exam for Data Structures, Algorithms and Databases

## Question 1 Entity-Relationship modelling

(a) Entity-relationship model with annotation of multiplicities and annotations.



- “a company that provides household services such as cleaning, painting, repairs, etc.”
  - Entity:
    - “company”, “household services”
  - Relationship
    - “provides”
  - Hierarchical
    - “cleaning”, “painting”, and “repairs” are are special cases of a more general “household services” entity (“super-entity”)
- “The company receives orders from clients for household jobs, schedules the work from its pool of workers, and bills the clients for the work carried out. ”
  - New Entity:
    - “orders”, “clients”, “work”, I omit this entity in the entity-relationship, because work is a combination of “household jobs” and “workers”. In this way, the diagram is simplified.
  - Relationship
    - “receives”, “schedules”, “bills”
- “The charge depends on the type of service and hours involved for carrying it out. Regular clients may be offered a discount.”
  - Attribute:
    - “charge”, “hours”, “discount”. Treat “discount” as an attribute, because regular clients may or may not be offered a discount when they are charged.
- “If a job was not satisfactorily carried out, a repeat visit is provided free of charge, possibly by a different worker. Customers give a satisfaction rating for each job, which is recorded by the company to assess the performance record of the workers.”
  - Attribute:
    - “performance record”. It can be an attribute that is in the order table. Because we can easily set the final charge to be free if a customer gives an unsatisfaction rating.

(b)

1. stating how each relationship and subclass entity in the model should be treated.
  - The services table shall have an attribute called “hourly\_rate”, that can be used to calculate the charge in the orders table.
  - As stated, “Workers are specilising in each kind of work”, so we set “sid” as a foreign key, which is a **one-to-one relationship** between table “services” and “workers”.
  - Workers are contracted to do a specific maximum number of hours per week, so we treat max working hours as an **attribute** of the “workers” entity.
  - The company should have a table to record **clients’** information, and a table of **orders**, that takes the “cid”, “sid”, and “wid” as foreign keys, from the “clients”, “services”, and “workers” tables respectively. The relationship between the orders table and the other tables is a **one-to-many relationship**. Because a client, service, or worker can have multiple orders.

- The orders table should also have a column “**hours\_worked**” to record the hours a worker has worked and the hours a worker needs to work. So it will not let the worker work exceed the maximum work hours per week. And update it each week accordingly.
2. Represent the design with relational schemas for tables.
  3. Annotate the schemas with primary keys and the possibility of null attributes.

services (sid, name, hourly\_rate)

workers (wid, sid, first\_name, last\_name, max\_hours\_per\_week)

clients (cid, first\_name, last\_name, birthday°, email°, phone)

orders (oid, cid, sid, wid, scheduled\_date, completed\_date°, original\_charge, discount°, final\_charge, hours\_to\_work, hours\_worked, rating°)

(c) Write SQL “CREATE TABLE” statements for 2-3 tables. Include at least one table that incorporates a relationship. Include all the necessary constraints and “ON DELETE” actions.

```
CREATE TABLE services (
    sid INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    hourly_rate DECIMAL(10, 2) NOT NULL CHECK (hourly_rate > 0)
);
```

```
CREATE TABLE workers (
    wid INT PRIMARY KEY AUTO_INCREMENT,
    sid INT NOT NULL,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    max_hours_per_week DECIMAL(3, 1) NOT NULL,
    FOREIGN KEY (sid) REFERENCES services(sid) ON DELETE NO ACTION
);
```

```
CREATE TABLE orders (
    oid INT PRIMARY KEY AUTO_INCREMENT,
    cid INT NOT NULL,
    sid INT NOT NULL,
    wid INT NOT NULL,
    scheduled_date DATE NOT NULL,
    completed_date DATE,
    original_charge DECIMAL(10, 2) NOT NULL CHECK (original_charge > 0),
    discount DECIMAL(5, 2),
    final_charge DECIMAL(10, 2) CHECK (final_charge >= 0 OR final_charge IS NULL),
    hours_to_work DECIMAL(3, 1) NOT NULL,
    hours_worked DECIMAL(3, 1) NOT NULL DEFAULT 0,
    rating DECIMAL(2, 1),
    FOREIGN KEY (cid) REFERENCES clients(cid) ON DELETE NO ACTION,
    FOREIGN KEY (sid) REFERENCES services(sid) ON DELETE NO ACTION,
    FOREIGN KEY (wid) REFERENCES workers(wid) ON DELETE NO ACTION,
);
```

## Question 2 Relational Algebra

(a)

C	X	Y	Z	D	W
a	x	15	e	NULL	NULL
b	x	10	f	p	1
b	x	10	f	r	4
b	x	15	g	NULL	NULL
a	y	15	h	p	3

(b)

**Relational algebra expression:**

$$\pi_{name}(courses \bowtie (\pi_{cid,sid}(\sigma_{year=2023}lecturing) - \pi_{cid,sid}(\sigma_{year<2023}lecturing)))$$

**Steps in SQL:**

- Courses taught by staff in 2023:  
SELECT cid, sid FROM lecturing WHERE year = 2023;  
 $\pi_{cid,sid}(\sigma_{year=2023}lecturing)$
- Courses taught by staff before 2023:  
SELECT cid, sid FROM lecturing WHERE year < 2023;  
 $\pi_{cid,sid}(\sigma_{year<2023}lecturing)$
- Find the set difference between them, so we exclude the record that the same staff taught the same course before 2023 (We cannot ignore 'sid' here, because a course may be taught by staff before 2023, but another staff only started to teach that course in 2023, and we still need this record.):  
SELECT cid, sid FROM lecturing WHERE year = 2023  
EXCEPT  
SELECT cid, sid FROM lecturing WHERE year < 2023;  
 $\pi_{cid,sid}(\sigma_{year=2023}lecturing) - \pi_{cid,sid}(\sigma_{year<2023}lecturing)$
- Finally, we select the name of that course after 'NATURAL JOIN' the filter table with table:  
SELECT name  
FROM course  
NATURAL JOIN (  
SELECT cid, sid FROM lecturing WHERE year = 2023  
NATURAL JOIN course  
EXCEPT  
SELECT cid, sid FROM lecturing WHERE year < 2023  
);  
 $\pi_{name}(courses \bowtie (\pi_{cid,sid}(\sigma_{year=2023}lecturing) - \pi_{cid,sid}(\sigma_{year<2023}lecturing)))$

Do not write below this line

**Statement of good academic conduct**

By submitting this assignment, I understand that I am agreeing to the following statement of good academic conduct:

- I confirm that this assignment is **my own work** and I have not worked with others in preparing this assignment.
- I confirm this assignment was written by me and is in my own words, except for any materials from published or other sources which are clearly indicated and acknowledged as such by appropriate referencing.
- I confirm that this work is not copied from any other person's work (published or unpublished), web site, book or other source, and has not previously been submitted for assessment either at the University of Birmingham or elsewhere.
- I confirm that I have not asked, or paid, others to prepare any part of this work for me.
- I confirm that I have read and understood the University regulations on plagiarism (<https://intranet.birmingham.ac.uk/as/registry/policy/conduct/plagiarism/index.aspx>).