~~Quiz~~ Take-Home Exercise: Row Overhead

Total Score: /6

Printed Name:

Due: Wed, 23 Feb before class

Rules:

1. You MAY use any printed or handwritten notes.

2. You MAY ~~NOT~~ use a computer or any other electronic device.

3. You MAY ~~NOT~~ discuss this quiz with any other human.

Note: You are allowed to use any resources you want. In particular, you should use the command

```
1   select typname,typalign,typlen from pg_type;
```

to get information about each of the types used in the problems below.

The references in the lecture notes also contain SQL queries that will reorder the columns for you automatically, but I recommend trying to do the problems by hand first.

Problem 1: The `country` table in the pagila database is defined as follows:

```
1  CREATE TABLE country (
2      country_id integer DEFAULT nextval('public.country_country_id_seq'::regclass)
3          NOT NULL,
4      country text NOT NULL,
5      last_update timestamp with time zone DEFAULT now() NOT NULL
6  );
```

Create a new, equivalent table whose column-order minimizes the amount of row overhead.

Problem 2: The `staff` table in the pagila database is defined as follows:

```
1  CREATE TABLE staff (
2      staff_id integer DEFAULT nextval('public.staff_staff_id_seq'::regclass) NOT NULL,
3      first_name text NOT NULL,
4      last_name text NOT NULL,
5      address_id integer NOT NULL,
6      email text,
7      store_id integer NOT NULL,
8      active boolean DEFAULT true NOT NULL,
9      username text NOT NULL,
10     password text,
11     last_update timestamp with time zone DEFAULT now() NOT NULL,
12     picture bytea
13 );
```

Create a new, equivalent table whose column-order minimizes the amount of row overhead.

Problem 3: The `film` table in the pagila database is defined as follows:

```
1  CREATE TABLE public.film (
2      film_id integer DEFAULT nextval('public.film_film_id_seq'::regclass) NOT NULL,
3      title text NOT NULL,
4      description text,
5      release_year public.year,
6      language_id integer NOT NULL,
7      original_language_id integer,
8      rental_duration smallint DEFAULT 3 NOT NULL,
9      rental_rate numeric(4,2) DEFAULT 4.99 NOT NULL,
10     length smallint,
11     replacement_cost numeric(5,2) DEFAULT 19.99 NOT NULL,
12     rating public.mpaa_rating DEFAULT 'G'::public.mpaa_rating,
13     last_update timestamp with time zone DEFAULT now() NOT NULL,
14     special_features text[],
15     fulltext tsvector NOT NULL
16  );
```

Create a new, equivalent table whose column-order minimizes the amount of row overhead.

NOTE: The `rental_rate` and `replacement_cost` fields are stored using the `numeric` type instead of `float`. `numeric` is an exact representation of a number, so it requires more computation and more storage, but it will never have rounding errors. This is the correct way to represent money in a SQL database.