


| | | | | |
|---|-------------------|--|--------|-------------|
|  | Lehrveranstaltung | Databases and Information Systems 2020 | | |
| | Aufgabenzettel | 1 | | |
| | STiNE-Gruppe 14 | Simon Weidmann, Aram Yesildeniz | | |
| | Ausgabe | 28. April 2020 | Abgabe | 8. Mai 2020 |

1.2 DB-Schema

Tabellen

=> Horizontal Partitioning

| Estate Agent | | | |
|--------------|------|---------|----------|
| <u>Login</u> | Name | Address | Password |
| | | | |

| Estate | | | | | | |
|-----------|------|-------------|--------|---------------|-------------|-----------------------------|
| <u>ID</u> | City | Postal Code | Street | Street Number | Square Area | Manager = EstateAgent.Login |
| | | | | | | |

| Apartment | | | | | | | | | | | |
|-----------|------|----|------|---------|----|-------|------|-------|---------|---------|------------------------------------|
| <u>ID</u> | City | PC | Str. | Str.Nr. | SA | Floor | Rent | Rooms | Balcony | Kitchen | <u>Manager = EstateAgent.Login</u> |
| | | | | | | | | | | | |

| House | | | | | | | | | |
|-----------|------|----|------|---------|----|--------|-------|--------|-----------------------------|
| <u>ID</u> | City | PC | Str. | Str.Nr. | SA | Floors | Price | Garden | Manager = EstateAgent.Login |
| | | | | | | | | | |

| Person | | | |
|-----------|------------|------|---------|
| <u>ID</u> | First Name | Name | Address |
| | | | |

| Contract | | |
|---------------------|------|-------|
| <u>Contract No.</u> | Date | Place |
| | | |


| Tenancy Contract | | | | | | | |
|------------------|------|-------|----------|------|-----------|-----------|--------------|
| <u>Con.No.</u> | Date | Place | Start D. | Dur. | Add.Costs | Person.ID | Apartment.ID |
| | | | | | | | |

| Purchase Contract | | | | | | |
|-------------------|------|-------|----------------|--------------|-----------|----------|
| <u>Con.No.</u> | Date | Place | No.of.Install. | Intrest Rate | Person.ID | House.ID |
| | | | | | | |

SQL Scripts

Datenbank erstellen

```
CREATE DATABASE dis
WITH
OWNER = postgres
ENCODING = 'UTF8'
CONNECTION LIMIT = -1;
```

| | | | | |
|---|-------------------|--|--------|-------------|
|  | Lehrveranstaltung | Databases and Information Systems 2020 | | |
| | Aufgabenzettel | 1 | | |
| | STiNE-Gruppe 14 | Simon Weidmann, Aram Yesildeniz | | |
| | Ausgabe | 28. April 2020 | Abgabe | 8. Mai 2020 |

Tabellen erstellen


```
CREATE TABLE public.estate_agent
(
    agent_login text,
    agent_name text,
    agent_address text,
    agent_password text,
    PRIMARY KEY (agent_login)
);
```

```
ALTER TABLE public.estate_agent
    OWNER to postgres;
```

```
CREATE TABLE public.estate
(
    estate_id serial,
    city text,
    postcal_code integer,
    street text,
    street_number text,
    square_area integer,
    manager text,
    PRIMARY KEY (estate_id),
    CONSTRAINT manager FOREIGN KEY (manager)
        REFERENCES public.estate_agent (agent_login) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
);
```

```
ALTER TABLE public.estate
    OWNER to postgres;
```

```
CREATE TABLE public.apartment
(
    floor integer,
    rent text,
    rooms text,
    balcony boolean,
    kitchen boolean,
```

| | | | | |
|---|-------------------|---|--------|--------------------|
|  | Lehrveranstaltung | Databases and Information Systems 2020 | | |
| | Aufgabenzettel | 1 | | |
| | STiNE-Gruppe 14 | Simon Weidmann, Aram Yesildeniz | | |
| | Ausgabe | 28. April 2020 | Abgabe | 8. Mai 2020 |

```

CONSTRAINT apartment_pkey PRIMARY KEY (estate_id),
CONSTRAINT manager FOREIGN KEY (manager)
    REFERENCES public.estate_agent (agent_login) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID
)
INHERITS (public.estate);

ALTER TABLE public.apartment
    OWNER to postgres;


CREATE TABLE public.house
(
    floors integer,
    price text,
    garden boolean,
    CONSTRAINT house_pkey PRIMARY KEY (estate_id),
    CONSTRAINT manager FOREIGN KEY (manager)
        REFERENCES public.estate_agent (agent_login) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
INHERITS (public.estate)
TABLESPACE pg_default;

ALTER TABLE public.house
    OWNER to postgres;

CREATE TABLE public.person
(
    id serial,
    first_name text,
    last_name text,
    address text,
    PRIMARY KEY (id)
);

ALTER TABLE public.person
    OWNER to postgres;

```

| | | | | |
|---|-------------------|--|--------|-------------|
|  | Lehrveranstaltung | Databases and Information Systems 2020 | | |
| | Aufgabenzettel | 1 | | |
| | STiNE-Gruppe 14 | Simon Weidmann, Aram Yesildeniz | | |
| | Ausgabe | 28. April 2020 | Abgabe | 8. Mai 2020 |


```
CREATE TABLE public.contract
(
    contract_number serial,
    contract_date date,
    place text,
    PRIMARY KEY (contract_number)
);
```

```
ALTER TABLE public.contract
    OWNER to postgres;
```

```
CREATE TABLE public.tenancy_contract
(
    start_date date,
    duration text,
    additional_costs text,
    person_id integer,
    apartment_id integer,
    CONSTRAINT tenancy_contract_pkey PRIMARY KEY (contract_number),
    CONSTRAINT person_id FOREIGN KEY (person_id)
        REFERENCES public.person (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT apartment_id FOREIGN KEY (apartment_id)
        REFERENCES public.apartment (estate_id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)
INHERITS (public.contract);
```

```
ALTER TABLE public.tenancy_contract
    OWNER to postgres;
```

```
CREATE TABLE public.purchase_contract
(
    installment_amount text,
    interest_rate text,
    person_id integer,
    house_id integer,
    CONSTRAINT purchase_contract_pkey PRIMARY KEY (contract_number),
```

| | | | | |
|---|-------------------|--|--------|-------------|
|  | Lehrveranstaltung | Databases and Information Systems 2020 | | |
| | Aufgabenzettel | 1 | | |
| | STiNE-Gruppe 14 | Simon Weidmann, Aram Yesildeniz | | |
| | Ausgabe | 28. April 2020 | Abgabe | 8. Mai 2020 |

```

CONSTRAINT person_id FOREIGN KEY (person_id)
  REFERENCES public.person (id) MATCH SIMPLE
  ON UPDATE NO ACTION
  ON DELETE NO ACTION
  NOT VALID,
CONSTRAINT house_id FOREIGN KEY (house_id)
  REFERENCES public.house (estate_id) MATCH SIMPLE
  ON UPDATE NO ACTION
  ON DELETE NO ACTION
  NOT VALID
)
INHERITS (public.contract);

ALTER TABLE public.purchase_contract
  OWNER to postgres;

```

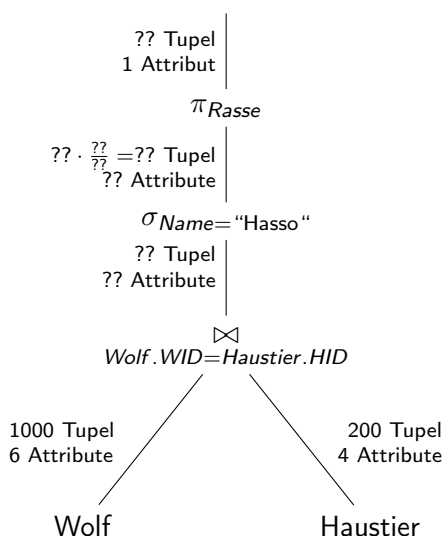
Estate Agent hinzufügen


```

INSERT INTO public.estate_agent(
  agent_login, agent_name, agent_address, agent_password)
VALUES ('testagent', 'testname', 'testaddress', 'testpassword');

```

1 Beispiel für Operatorbaum



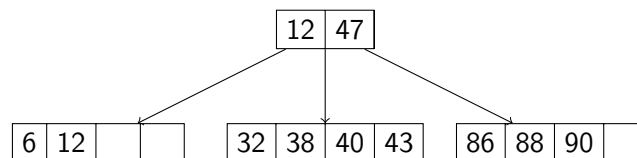
| | | | | |
|---|-------------------|--|--------|-------------|
|  | Lehrveranstaltung | Databases and Information Systems 2020 | | |
| | Aufgabenzettel | 1 | | |
| | STiNE-Gruppe 14 | Simon Weidmann, Aram Yesildeniz | | |
| | Ausgabe | 28. April 2020 | Abgabe | 8. Mai 2020 |

2 Beispiel fÄijrr Tabelle mit Sperranforderungen

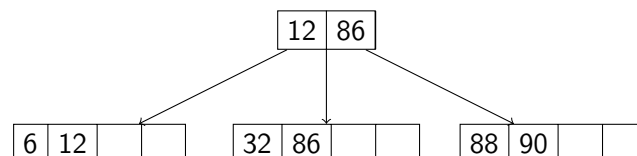
| Zeitschritt | T ₁ | T ₂ | T ₃ | x | y | z | Bemerkung |
|-------------|----------------|----------------|----------------|----------------|----------------|----|-----------|
| 0 | | | | NL | NL | NL | |
| 1 | lock(x,X) | | | X ₁ | NL | NL | |
| 2 | write(x) | lock(y,R) | | X ₁ | R ₂ | NL | |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |

3 Beispiel fÄijr B- und B*-BÄdumen

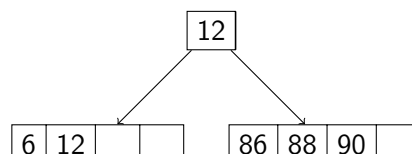
LÄuschen Sie aus dem unten abgebildeten **B*-Baum** der Klasse $\tau(1, 2, h)$ die DatensÄdtze mit den SchlÄijsseln **40, 43, 38, 32** und **90** (in dieser Reihenfolge). Geben Sie jeweils kurz an, welche konkrete MaÄßnahme Sie durchgefÄijhrt haben (Mischen, Ausgleichen, einfaches LÄuschen) und zeichnen Sie den Baum nach jedem Mischen und Ausgleichen neu. FÄijr Ausgleichs- und Mischoperationen sollen nur direkt benachbarte Geschwisterknoten (bevorzugt der rechte) herangezogen werden.



40 und 43, Einfaches LÄuschen
38, Ausgleichen



32, Mischen



90, Einfaches LÄuschen