SARA'S DAYCARE CENTER

Database Design by Sara Ogorzalek

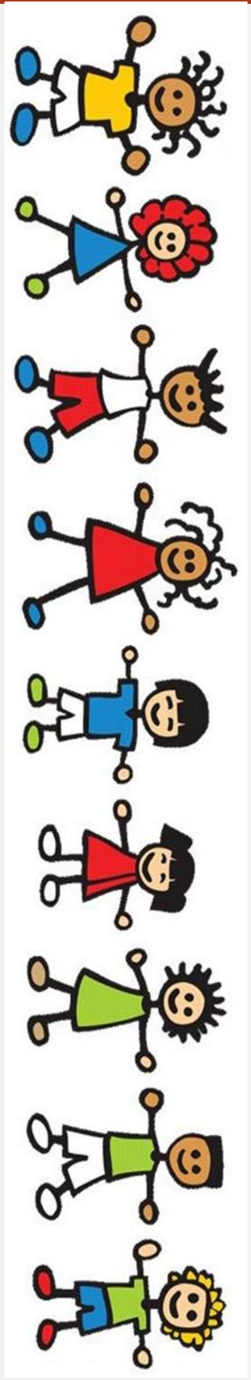
Table of Contents

Executive Summary

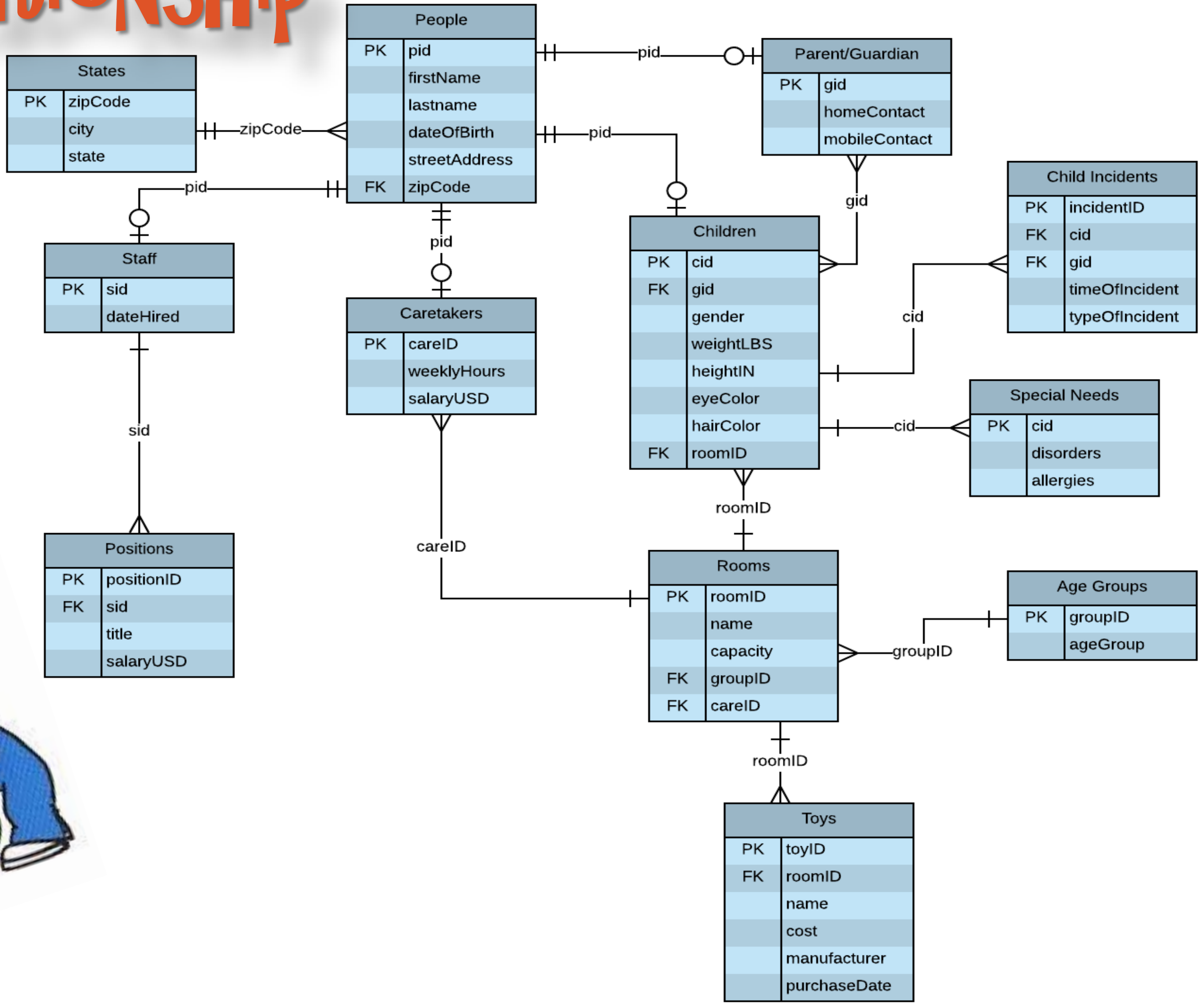
Located in Poughkeepsie, NY, Sara's Daycare Center is a small center where Database professionals can drop off their kids when they go to work. In the Daycare Center:

- *There are 3 age groups – Toddler, Early Preschool, and Preschool.*
- *Each age group belongs to a room.*
- *In each room, there is typically one caretaker however a room can have many caretakers.*
- *Children in the daycare are required to have at least one legal guardian in the system, but both guardians can be in the system as well.*
- *Incidents such as an injury are recorded into the database, as well as any special needs the child may have such as a food allergy.*

The database oversees all the people who work in the daycare, and keeps track of the children and the rooms. This paper outlines a database designed in Postgres needed to keep track of Sara's Daycare Center.



Entity-Relationship Diagram



Tables

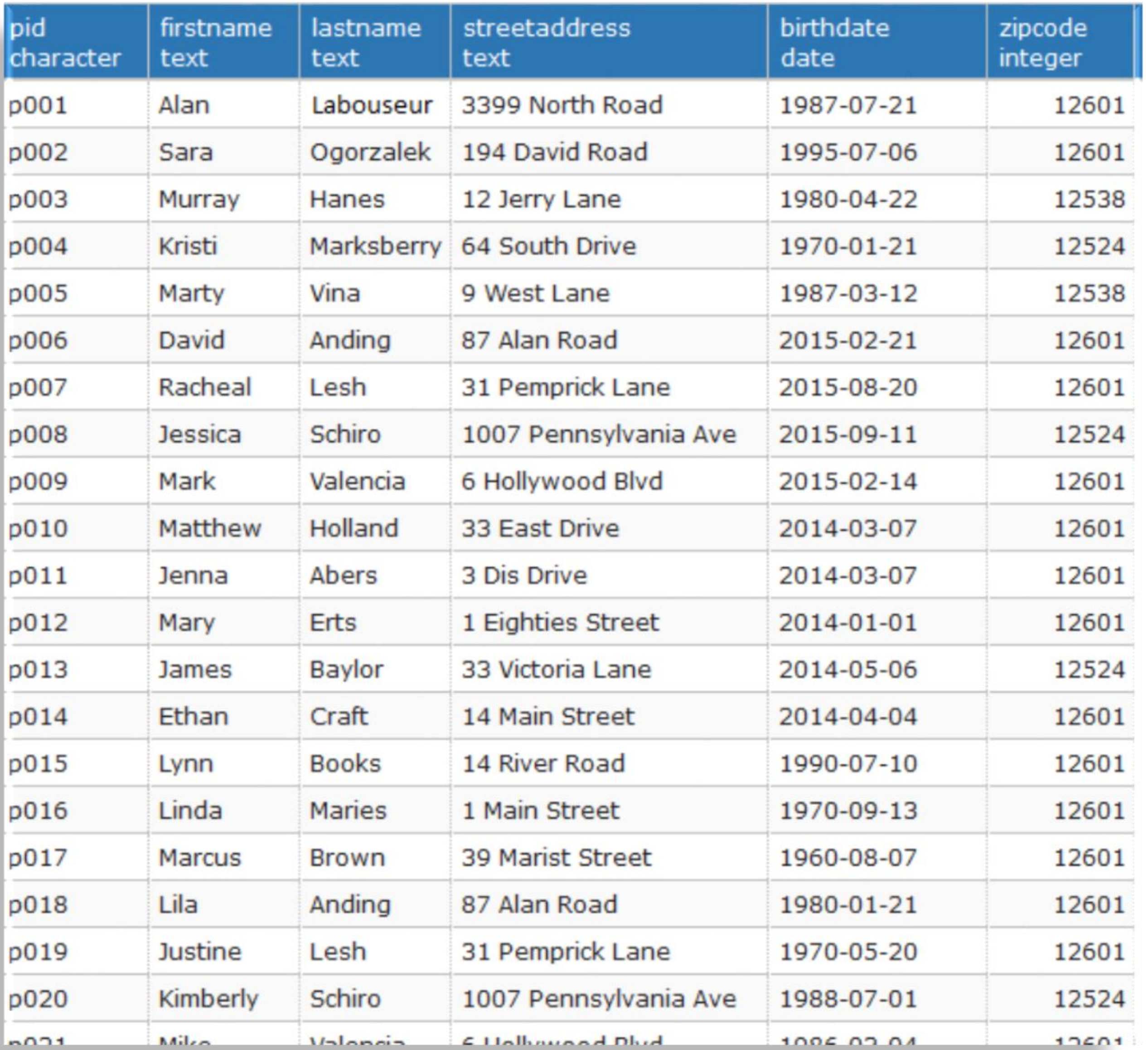
PEOPLE – the people table lists all the people in the daycare center

```
CREATE TABLE people (
    pid              CHAR(6) not null,
    firstName        text,
    lastName         text,
    streetAddress    text,
    birthDate        date not null,
    zipCode          integer not null,
    primary key(pid),
    foreign key(zipCode) references states(zipCode)
    );
```

functional dependencies

pid → firstName, lastName, streetAddress, birthDate, zipCode

Tables Sample PEOPLE data

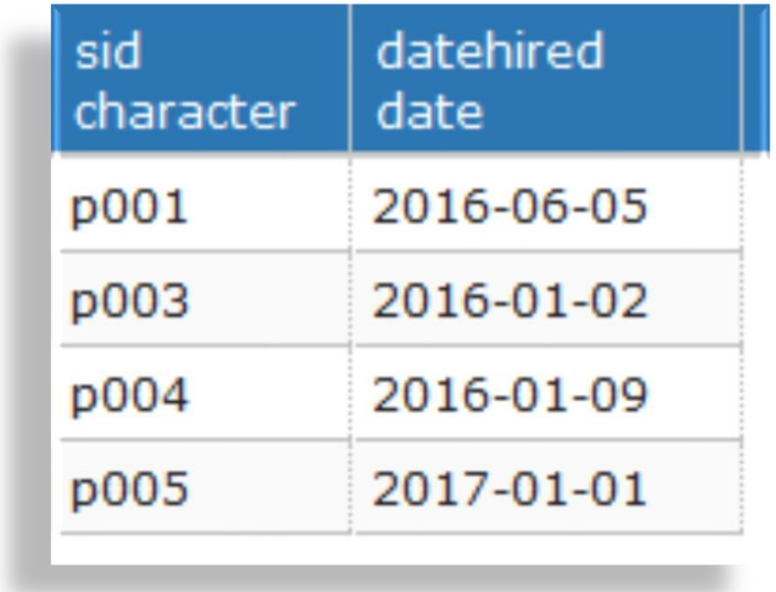


pid character	firstname text	lastname text	streetaddress text	birthdate date	zipcode integer
p001	Alan	Labouseur	3399 North Road	1987-07-21	12601
p002	Sara	Ogorzalek	194 David Road	1995-07-06	12601
p003	Murray	Hanes	12 Jerry Lane	1980-04-22	12538
p004	Kristi	Marksberry	64 South Drive	1970-01-21	12524
p005	Marty	Vina	9 West Lane	1987-03-12	12538
p006	David	Anding	87 Alan Road	2015-02-21	12601
p007	Racheal	Lesh	31 Pemprick Lane	2015-08-20	12601
p008	Jessica	Schiro	1007 Pennsylvania Ave	2015-09-11	12524
p009	Mark	Valencia	6 Hollywood Blvd	2015-02-14	12601
p010	Matthew	Holland	33 East Drive	2014-03-07	12601
p011	Jenna	Abers	3 Dis Drive	2014-03-07	12601
p012	Mary	Erts	1 Eighties Street	2014-01-01	12601
p013	James	Baylor	33 Victoria Lane	2014-05-06	12524
p014	Ethan	Craft	14 Main Street	2014-04-04	12601
p015	Lynn	Books	14 River Road	1990-07-10	12601
p016	Linda	Maries	1 Main Street	1970-09-13	12601
p017	Marcus	Brown	39 Marist Street	1960-08-07	12601
p018	Lila	Anding	87 Alan Road	1980-01-21	12601
p019	Justine	Lesh	31 Pemprick Lane	1970-05-20	12601
p020	Kimberly	Schiro	1007 Pennsylvania Ave	1988-07-01	12524
p021	Mike	Valencia	6 Hollywood Blvd	1986-02-04	12601

Tables

STAFF – the staff table lists all the staff in the daycare center

```
CREATE TABLE staff (
     sid              char(6) not null references people(pid),
     dateHired        date not null,
     primary key(sid)
     );
```



sid character	datehired date
p001	2016-06-05
p003	2016-01-02
p004	2016-01-09
p005	2017-01-01

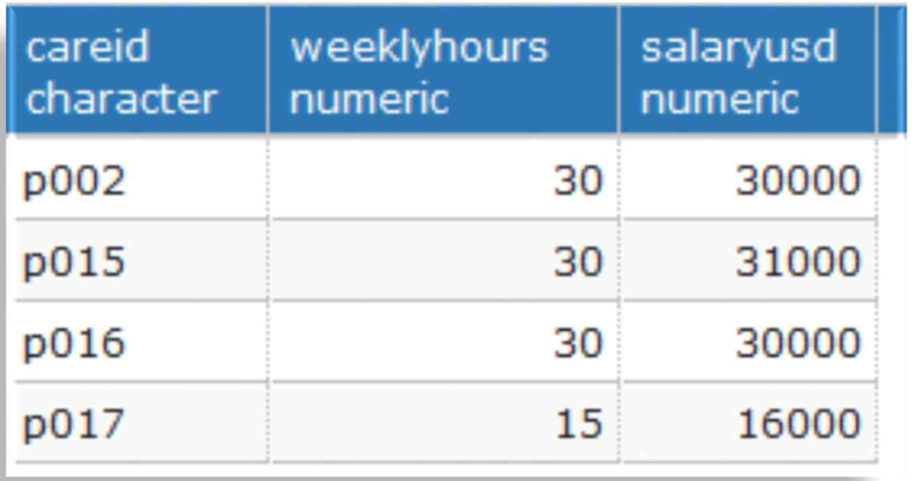
functional dependencies

sid → dateHired

Tables

CARETAKERS – the caretakers table lists all the caretakers in the daycare center

```
CREATE TABLE caretakers (
    careID          char(6) not null references people(pid),
    weeklyHours     numeric not null,
    salaryUSD       numeric not null,
    primary key(careID)
    );
```



careid character	weeklyhours numeric	salaryusd numeric
p002	30	30000
p015	30	31000
p016	30	30000
p017	15	16000

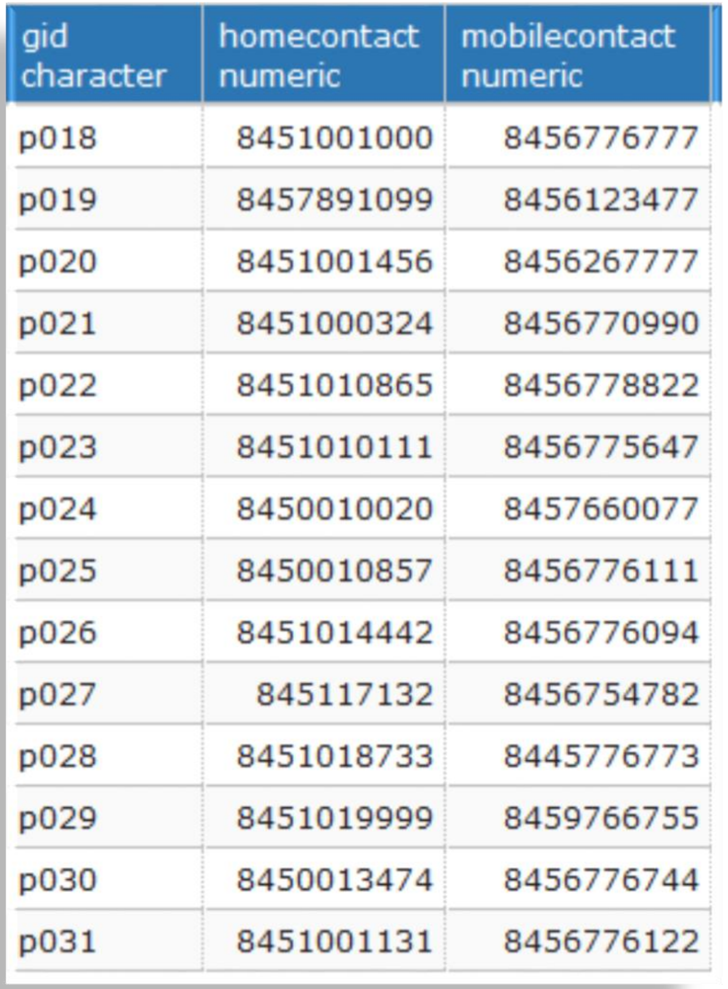
functional dependencies

careID → weeklyHours, salaryUSD

Tables

PARENT/GUARDIAN – the guardian table lists one or both of the guardians of the children in the daycare center

```
CREATE TABLE guardian (
    gid             char(4) not null references people(pid),
    homeContact     numeric not null,
    mobileContact   numeric not null,
    primary key(gid)
    );
```



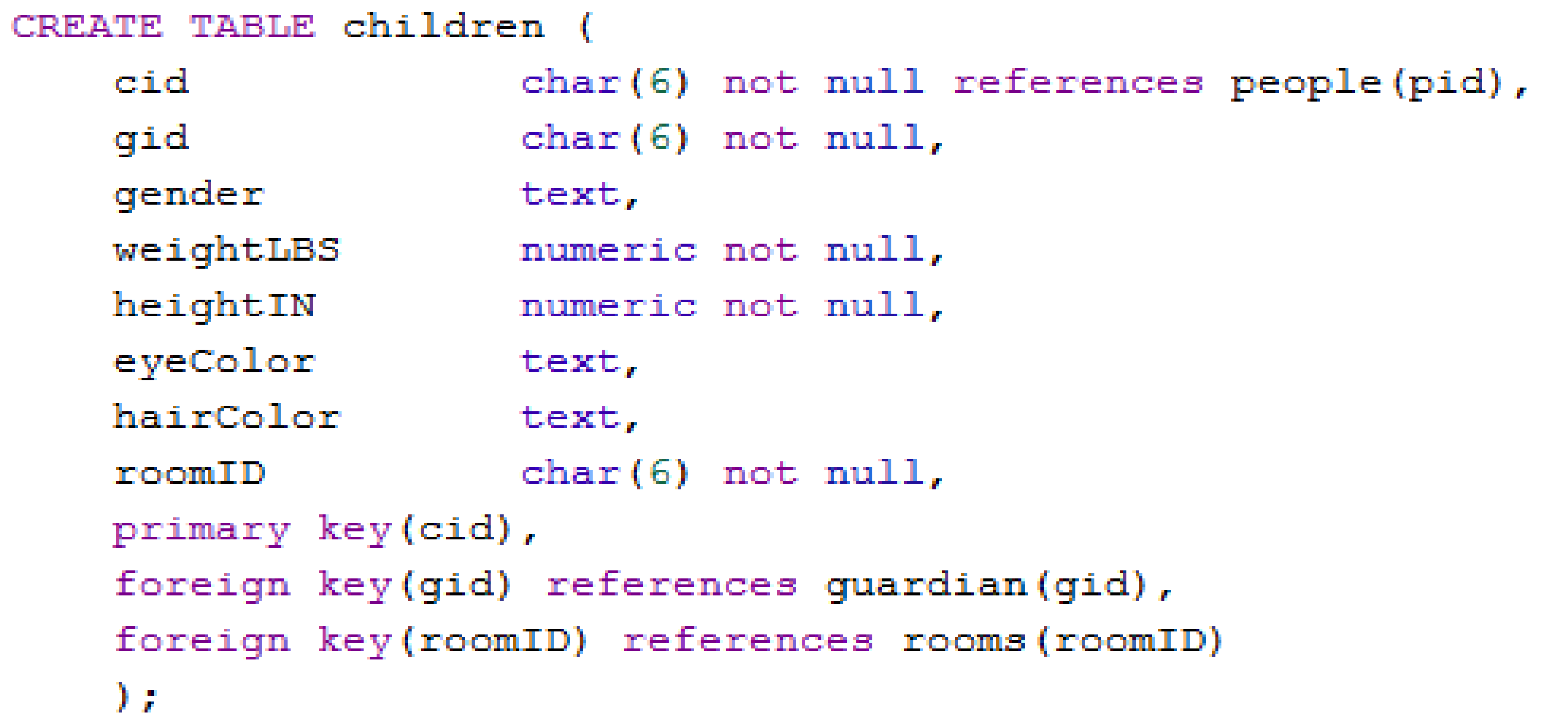
gid character	homecontact numeric	mobilecontact numeric
p018	8451001000	8456776777
p019	8457891099	8456123477
p020	8451001456	8456267777
p021	8451000324	8456770990
p022	8451010865	8456778822
p023	8451010111	8456775647
p024	8450010020	8457660077
p025	8450010857	8456776111
p026	8451014442	8456776094
p027	845117132	8456754782
p028	8451018733	8445776773
p029	8451019999	8459766755
p030	8450013474	8456776744
p031	8451001131	8456776122

functional dependencies

gid → homeContact, mobileContact

Tables

CHILDREN – the children table lists all the children in the daycare center



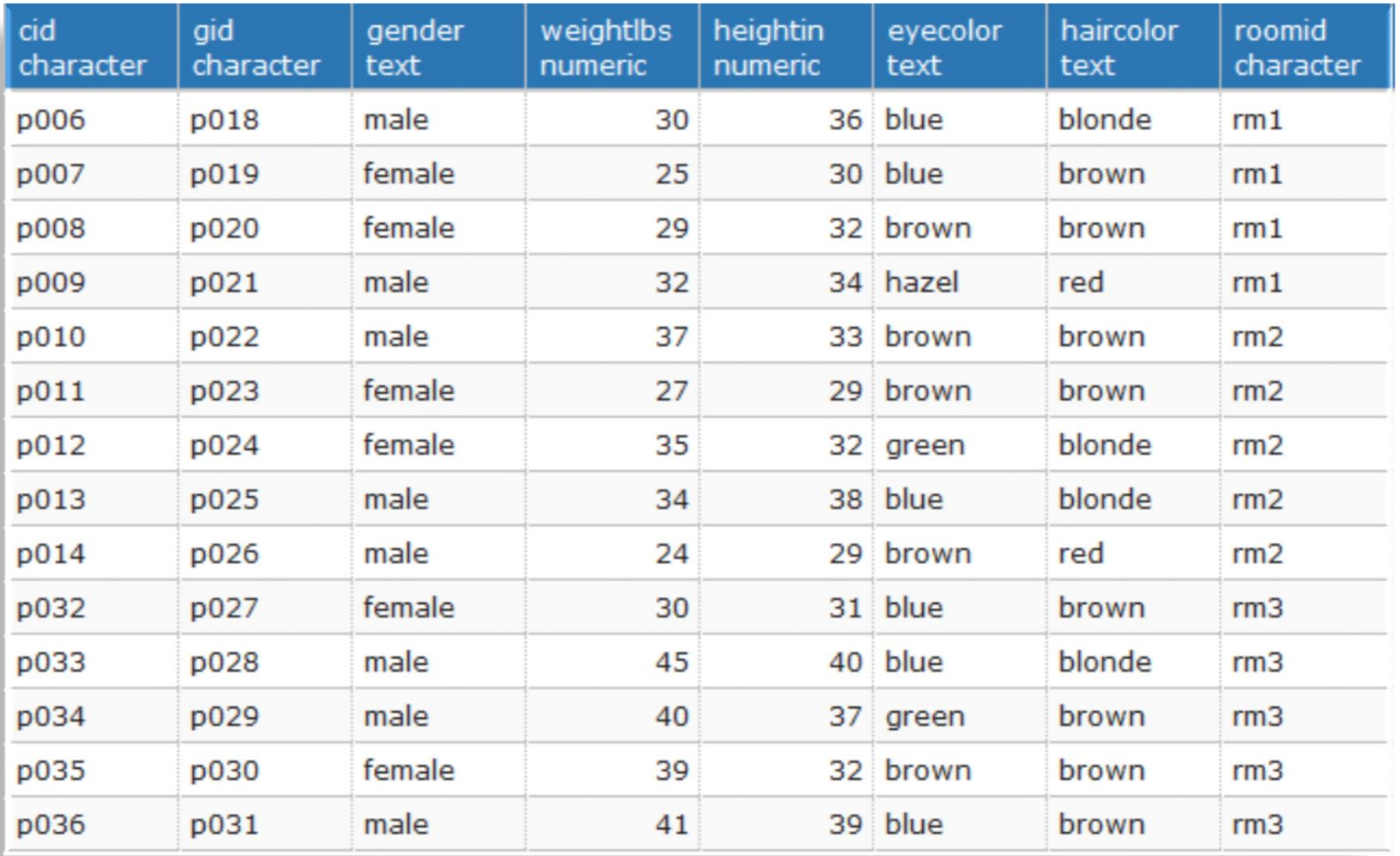
```
CREATE TABLE children (
    cid             char(6) not null references people(pid),
    gid             char(6) not null,
    gender          text,
    weightLBS       numeric not null,
    heightIN        numeric not null,
    eyeColor        text,
    hairColor       text,
    roomID          char(6) not null,
    primary key(cid),
    foreign key(gid) references guardian(gid),
    foreign key(roomID) references rooms(roomID)
    );
```

functional dependencies

cid → gender, weightLBS, heightIN, eyeColor, hairColor, roomID, gid

Tables

Sample CHILDREN data



cid character	gid character	gender text	weightlbs numeric	heightin numeric	eyecolor text	haircolor text	roomid character
p006	p018	male	30	36	blue	blonde	rm1
p007	p019	female	25	30	blue	brown	rm1
p008	p020	female	29	32	brown	brown	rm1
p009	p021	male	32	34	hazel	red	rm1
p010	p022	male	37	33	brown	brown	rm2
p011	p023	female	27	29	brown	brown	rm2
p012	p024	female	35	32	green	blonde	rm2
p013	p025	male	34	38	blue	blonde	rm2
p014	p026	male	24	29	brown	red	rm2
p032	p027	female	30	31	blue	brown	rm3
p033	p028	male	45	40	blue	blonde	rm3
p034	p029	male	40	37	green	brown	rm3
p035	p030	female	39	32	brown	brown	rm3
p036	p031	male	41	39	blue	brown	rm3

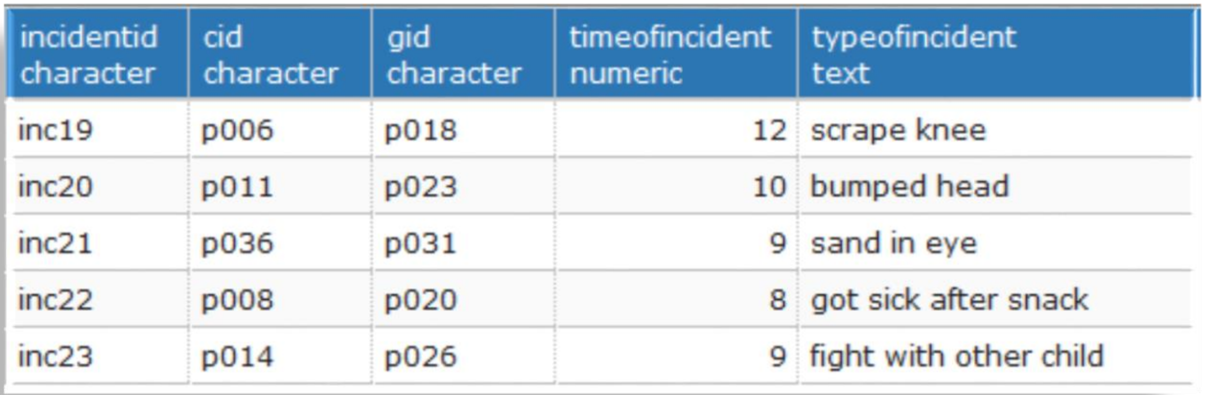
Tables

CHILD INCIDENTS – the incidents table lists all the incidents in the daycare center such as a fall or injury

```
CREATE TABLE incidents (
    incidentID      char(6) not null,
    cid             char(6) not null,
    gid             char(6) not null,
    timeOfIncident  numeric not null,
    typeOfIncident  text,
    primary key(incidentID),
    foreign key(cid) references children(cid),
    foreign key(gid) references guardian(gid)
    );
```

functional dependencies

incidentID → cid, gid, timeOfIncident, typeOfIncident

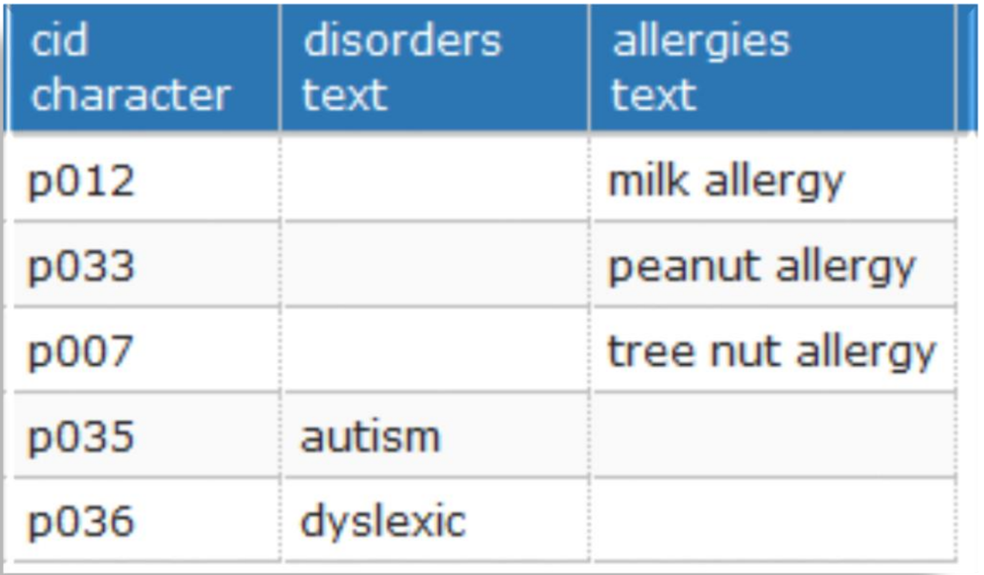


incidentid character	cid character	gid character	timeofincident numeric	typeofincident text
inc19	p006	p018	12	scrape knee
inc20	p011	p023	10	bumped head
inc21	p036	p031	9	sand in eye
inc22	p008	p020	8	got sick after snack
inc23	p014	p026	9	fight with other child

Tables

SPECIAL NEEDS – the special needs table lists all the special needs for the children in the daycare center including allergies

```
CREATE TABLE specialneeds (
    cid            char(6) not null references children(cid),
    disorders      text,
    allergies      text,
    primary key(cid)
    );
```



cid character	disorders text	allergies text
p012		milk allergy
p033		peanut allergy
p007		tree nut allergy
p035	autism	
p036	dyslexic	

functional dependencies
cid → disorders, allergies

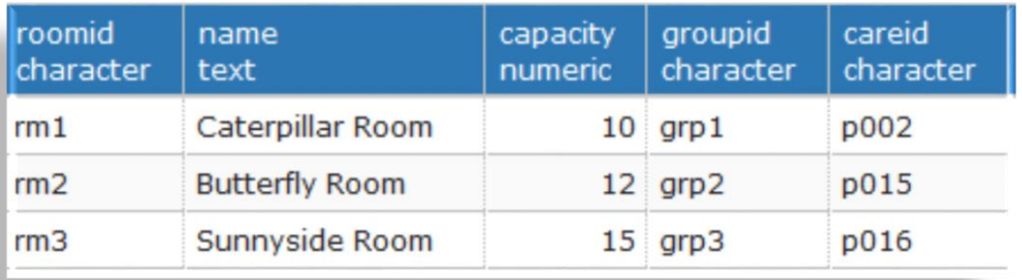
Tables

ROOMS – the rooms table lists all the rooms for children in the daycare center

```
CREATE TABLE rooms (
     roomID             char(4) not null,
     name               text,
     capacity           numeric not null,
     groupID            char(6) not null,
     careID             char(6) not null,
     primary key(roomID),
     foreign key(groupID) references agegroup(groupID),
     foreign key(careID) references caretakers(careID)
     );
```

functional dependencies

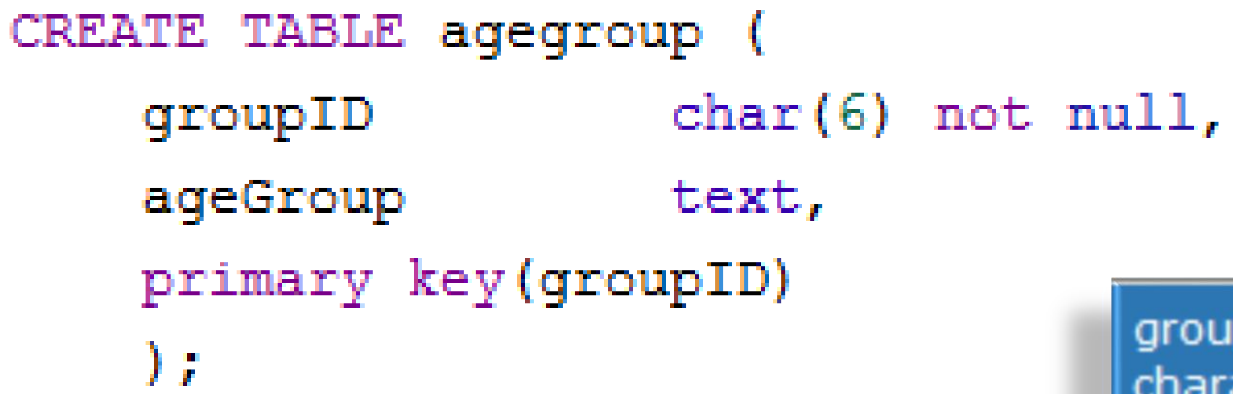
roomID → name, capacity, groupID, careID



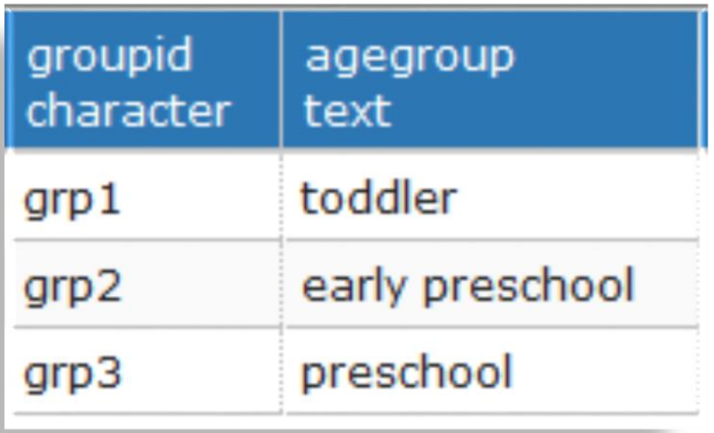
roomid character	name text	capacity numeric	groupid character	careid character
rm1	Caterpillar Room	10	grp1	p002
rm2	Butterfly Room	12	grp2	p015
rm3	Sunnyside Room	15	grp3	p016

Tables

AGE GROUPS – the age group table lists all the age groups in the daycare center which are placed in rooms.



```
CREATE TABLE agegroup (
    groupID         char(6) not null,
    ageGroup        text,
    primary key(groupID)
    );
```



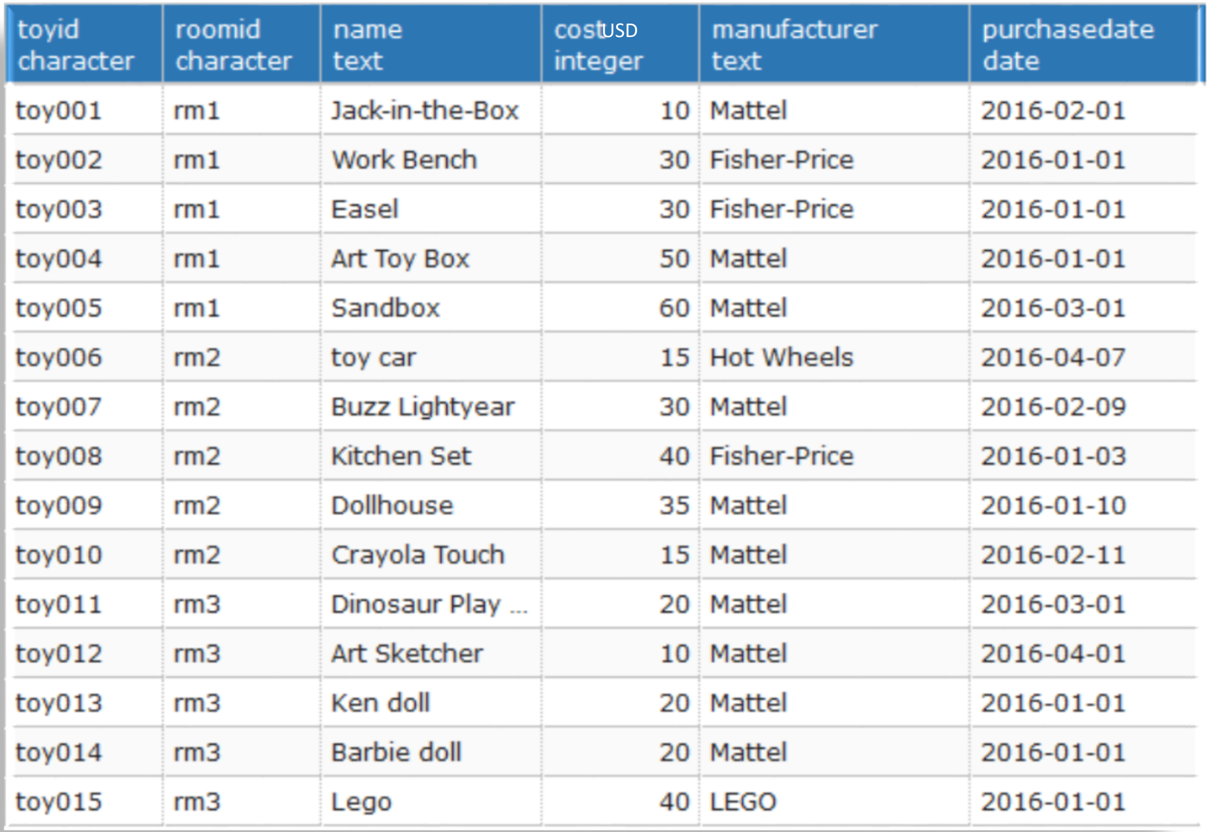
groupid character	agegroup text
grp1	toddler
grp2	early preschool
grp3	preschool

functional dependencies
groupID → ageGroup

Tables

TOYS– the toys table lists all the toys in the daycare center based by room. It is important to keep track of toys to know which ones are getting old and costs.

```
CREATE TABLE toys (
    toyID            char(6) not null,
    roomID           char(6) not null,
    name             text,
    costUSD          integer not null,
    manufacturer     text,
    purchaseDate     date not null,
    primary key(toyID),
    foreign key(roomID) references rooms(roomID)
    );
```



toyid character	roomid character	name text	costUSD integer	manufacturer text	purchasedate date
toy001	rm1	Jack-in-the-Box	10	Mattel	2016-02-01
toy002	rm1	Work Bench	30	Fisher-Price	2016-01-01
toy003	rm1	Easel	30	Fisher-Price	2016-01-01
toy004	rm1	Art Toy Box	50	Mattel	2016-01-01
toy005	rm1	Sandbox	60	Mattel	2016-03-01
toy006	rm2	toy car	15	Hot Wheels	2016-04-07
toy007	rm2	Buzz Lightyear	30	Mattel	2016-02-09
toy008	rm2	Kitchen Set	40	Fisher-Price	2016-01-03
toy009	rm2	Dollhouse	35	Mattel	2016-01-10
toy010	rm2	Crayola Touch	15	Mattel	2016-02-11
toy011	rm3	Dinosaur Play ...	20	Mattel	2016-03-01
toy012	rm3	Art Sketcher	10	Mattel	2016-04-01
toy013	rm3	Ken doll	20	Mattel	2016-01-01
toy014	rm3	Barbie doll	20	Mattel	2016-01-01
toy015	rm3	Lego	40	LEGO	2016-01-01

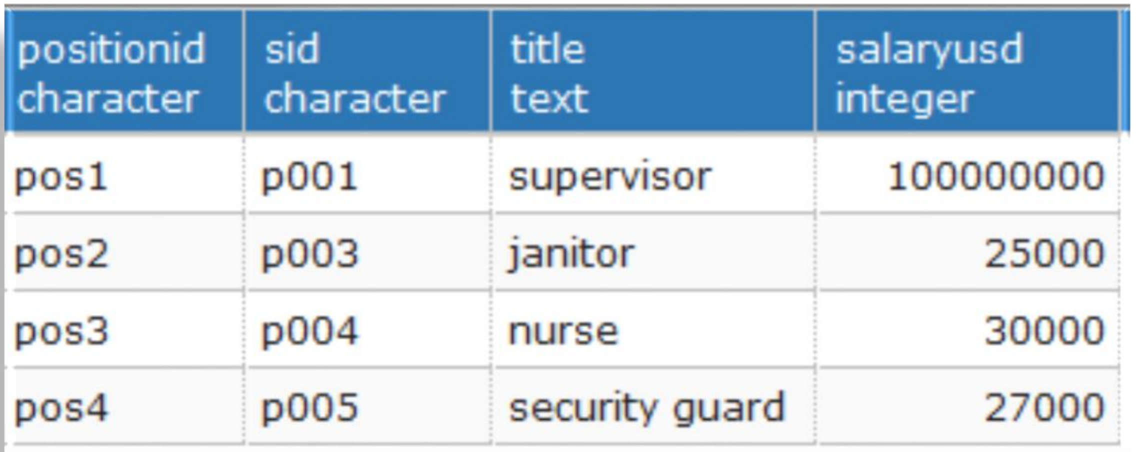
functional dependencies

toyID → roomID, name, cost, manufacturer, purchaseDate

Tables

POSITIONS– the positions table lists all the staff positions in the daycare center.

```
CREATE TABLE positions (
    positionID      char(6) not null,
    sid             char(6) not null,
    title           text,
    salaryUSD       integer not null,
    primary key(positionID),
    foreign key(sid) references staff(sid)
    );
```



positionid character	sid character	title text	salaryusd integer
pos1	p001	supervisor	100000000
pos2	p003	janitor	25000
pos3	p004	nurse	30000
pos4	p005	security guard	27000

*

*p001 is Alan

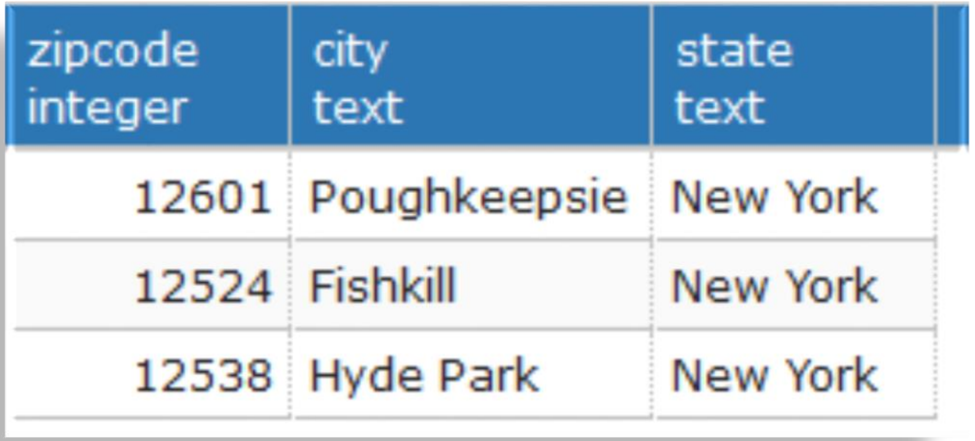
functional dependencies

toyID → roomID, name, cost, manufacturer, purchaseDate

Tables

STATES– the states table lists the city and state depending on zip code in the daycare center. These are primarily around Poughkeepsie, where the daycare is located.

```
CREATE TABLE states (
    zipCode      integer not null unique,
    city         text not null,
    state        text not null,
    primary key(zipCode)
    );
```



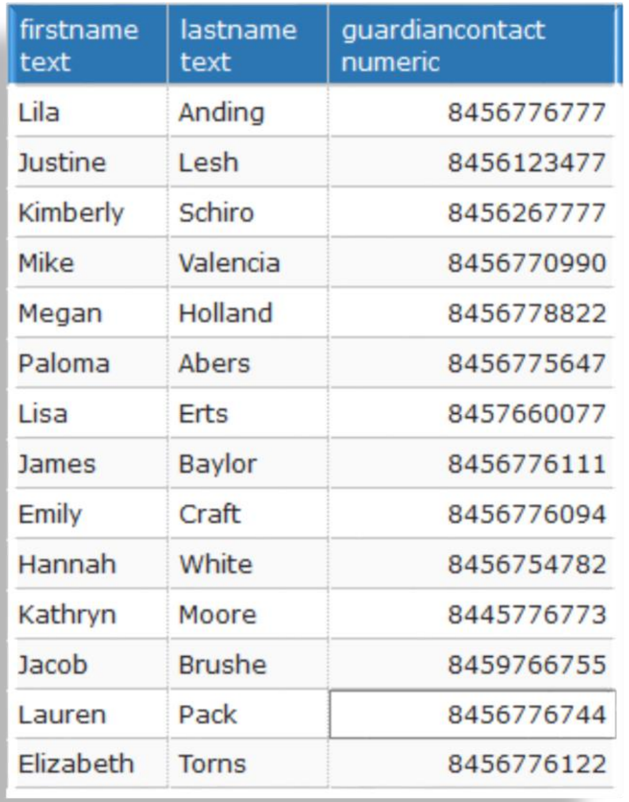
zipcode integer	city text	state text
12601	Poughkeepsie	New York
12524	Fishkill	New York
12538	Hyde Park	New York

functional dependencies
zipCode → city, state

Views

ChildrenGuardian lists the names of the children their parents emergency contact number. This is a good view for caretakers needing a child’s emergency contact.

```
CREATE VIEW ChildrenGuardian AS
SELECT firstName, lastName, guardian.mobileContact as GuardianContact
FROM people
inner join guardian
ON people.pid = guardian.gid;
```

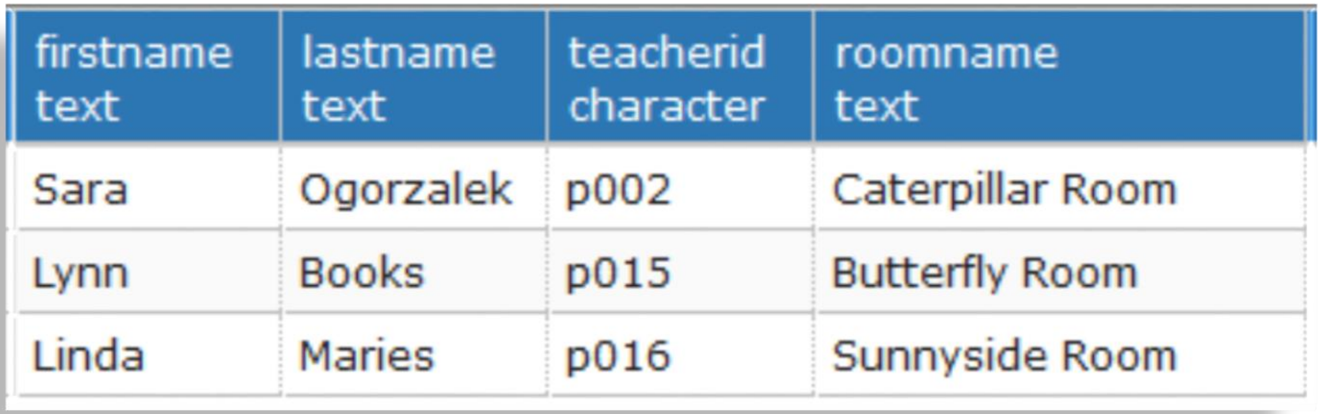


firstname text	lastname text	guardiancontact numeric
Lila	Anding	8456776777
Justine	Lesh	8456123477
Kimberly	Schiro	8456267777
Mike	Valencia	8456770990
Megan	Holland	8456778822
Paloma	Abers	8456775647
Lisa	Erts	8457660077
James	Baylor	8456776111
Emily	Craft	8456776094
Hannah	White	8456754782
Kathryn	Moore	8445776773
Jacob	Brushe	8459766755
Lauren	Pack	8456776744
Elizabeth	Torns	8456776122

Views

CaretakerRoom lists the first and last names of caretakers along with their ID number and room that they are assigned to. This is helpful to the supervisor and to parents.

```
CREATE VIEW CaretakerRoom AS
SELECT firstName, lastName, caretakers.careID as TeacherID, rooms.name as RoomName
FROM people
inner join caretakers
ON people.pid = caretakers.careID
inner join rooms
ON people.pid = rooms.careID;
```

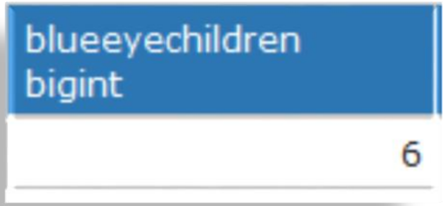


firstname text	lastname text	teacherid character	roomname text
Sara	Ogorzalek	p002	Caterpillar Room
Lynn	Books	p015	Butterfly Room
Linda	Maries	p016	Sunnyside Room

Reports

Query to return the number of children that are currently in the daycare who have blue eyes

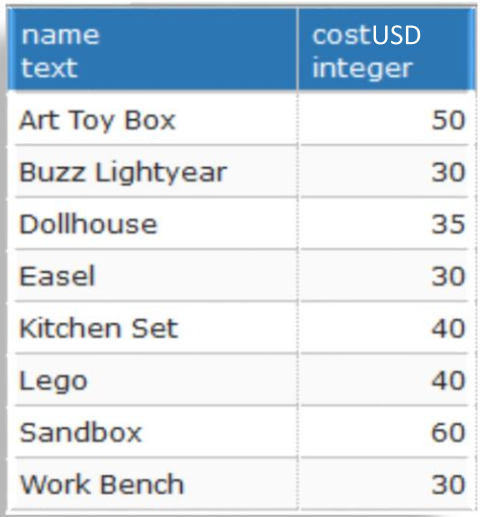
```
SELECT COUNT(pid) as blueEyeChildren
FROM people
inner join children
on people.pid = children.cid
WHERE eyeColor = 'blue';
```



blueeyechildren bigint
6

Query to display the names of toys whose costUSD is above the average costUSD in alphabetical order

```
SELECT t.name, t.cost
FROM toys t
where t.cost > (SELECT AVG(t.cost)
                FROM toys t
                )
ORDER BY t.name ASC;
```

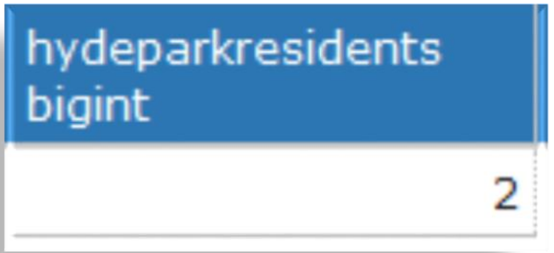


name text	costUSD integer
Art Toy Box	50
Buzz Lightyear	30
Dollhouse	35
Easel	30
Kitchen Set	40
Lego	40
Sandbox	60
Work Bench	30

Reports

Query to return the number of people in the database who live in Hyde Park

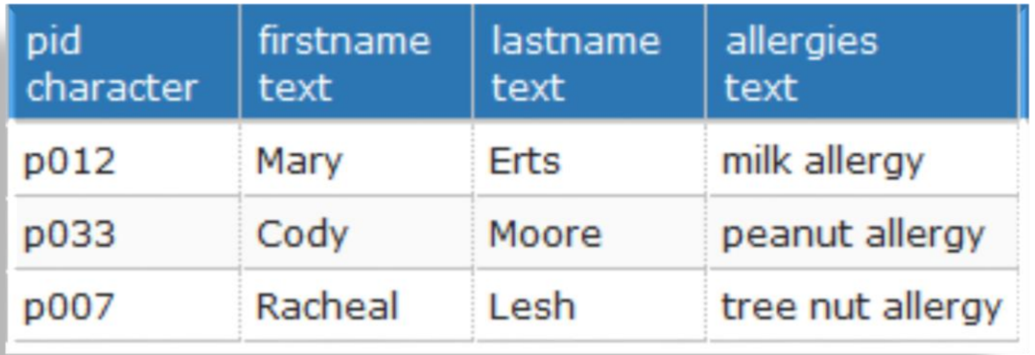
```sql
SELECT COUNT(pid) as HydeParkResidents
FROM people
inner join states
on states.zipCode = people.zipCode
WHERE city = 'Hyde Park';
```



hydeparkresidents bigint
2

Query to return the IDs and names of children who have a special need of an allergy and not a disorder

```sql
SELECT pid, firstName, lastName, specialneeds.allergies
FROM people
inner join specialneeds
ON people.pid = specialneeds.cid
WHERE specialneeds.allergies is not null;
```



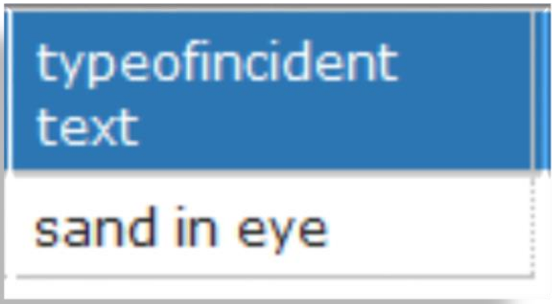
pid character	firstname text	lastname text	allergies text
p012	Mary	Erts	milk allergy
p033	Cody	Moore	peanut allergy
p007	Racheal	Lesh	tree nut allergy

Stored Procedures

IncidentsOf automatically returns a table of all the types of incidents that a child in the daycare has had when child ID is input. The daycare can use this information if parents want a history.

```
CREATE OR REPLACE FUNCTION IncidentsOf (char(6), REFCURSOR) returns refcursor as $$
DECLARE
            targetIncident  char(6) :=$1;
            resultset       REFCURSOR :=$2;
BEGIN
            open resultset for
            SELECT typeOfIncident
            FROM incidents
            WHERE cid = targetIncident;
        RETURN resultset;
END;
$$
language plpgsql;
```

```
select IncidentsOf('p036', 'results');
fetch all from results;
```



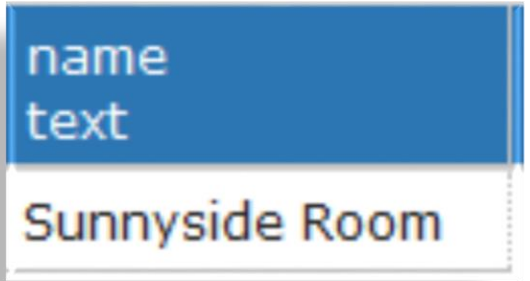
typeofincident text
sand in eye

Stored Procedures

InRoom automatically returns the name of the room that toy is in when the toy name is input. This can be helpful for knowing what toys belong to which rooms if one gets misplaced or lost. Some younger children cannot play with toys meant for older children because of a choking hazard.

```
CREATE OR REPLACE FUNCTION InRoom (text, REFCURSOR) returns refcursor as $$
DECLARE
            targetRoom       text :=$1;
            resultset        REFCURSOR :=$2;
BEGIN
            open resultset for
            SELECT rooms.name
            FROM rooms
            inner join toys
            ON rooms.roomID = toys.roomID
            WHERE toys.name = targetRoom;
        RETURN resultset;
END;
$$
language plpgsql;
```

```
select InRoom('Barbie doll', 'results');
fetch all from results;
```



name text
Sunnyside Room

Triggers

NewChild: This example triggers when a new entry is created for a new child

```
CREATE OR REPLACE FUNCTION NewChild()
RETURNS trigger AS
$$
BEGIN
    IF NEW.is_children = true THEN
        INSERT INTO children VALUES(NEW.cid, NEW.gid, NEW.gender, NEW.weightLBS,
                                    NEW.heightIN, NEW.eyeColor, NEW.hairColor, NEW.roomID);
    END IF;
    RETURN NEW;
END;
$$
language plpgsql;

CREATE TRIGGER addChild
AFTER INSERT OR UPDATE ON people
FOR EACH ROW
EXECUTE PROCEDURE NewChild();
```

Security

Administrator: The administrator has full access to all tables in the database

```
CREATE ROLE admin;
GRANT ALL ON ALL TABLES IN SCHEMA PUBLIC TO admin;
```

Staff: The staff have limited access to specific tables in the database

```
CREATE ROLE staff;
REVOKE ALL ON ALL TABLES IN SCHEMA PUBLIC FROM staff;
GRANT SELECT ON ALL TABLES IN SCHEMA PUBLIC TO staff;
GRANT INSERT ON  people, guardian, staff, incidents
TO staff;
GRANT UPDATE ON  people, guardian, staff, incidents
TO staff;
```

Security

Caretakers: Caretakers have limited access to some tables such as inserting a child incident

```
CREATE ROLE caretakers;
REVOKE ALL ON ALL TABLES IN SCHEMA PUBLIC FROM caretakers;
GRANT SELECT ON ALL TABLES IN SCHEMA PUBLIC TO caretakers;
GRANT INSERT ON people, guardian, incidents, specialneeds
TO caretakers;
GRANT UPDATE ON people, guardian, incidents, specialneeds
TO caretakers;
```

Implementation Notes & Known Problems

Overall, if this were a real daycare database filled with real data and not just sample data, there would be a lot more people in the building such as staff, children, and caretakers. There would have to be more rooms and several more of the same age group. For example, there would be two “Toddler” age groups.

Also, if this database were being implemented into an already established daycare, it would probably be unrealistic to keep track and know the cost of every toy. Unless they are new purchases immediately put into the database, it would be difficult to input every toy, which could be hundreds, into the database.

Future Enhancements

I would further enhance this database to be able to fit the needs of a larger and growing daycare. Enhancements I would make include adding a registration table to keep track of when parents registered for the daycare and if there are any parents on a waitlist. Another table I would add to the database is parent payment. This would keep track of the method the parent pays in and the amount. I would also have the database keep track of any kind of schedule the parents are on – for example if they only need to drop off their kid two times a week. This would enable an open spot for another parent's child with similar needs. The possibilities to enhance this daycare database are endless.