

eatemp.com



# HOSPITAL DATABASE

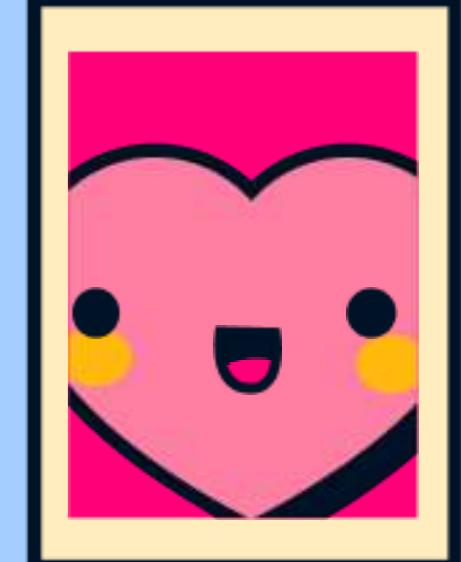
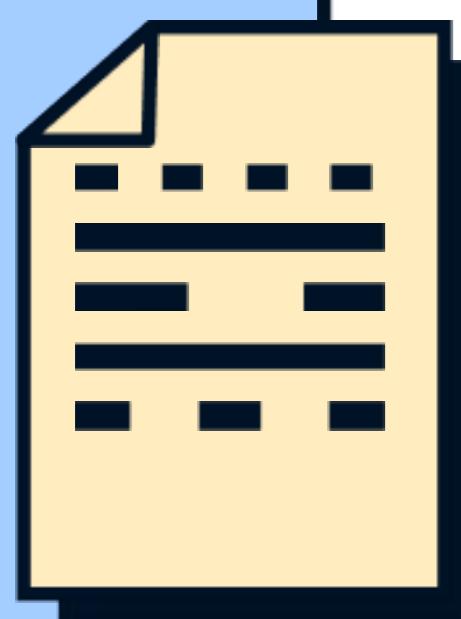
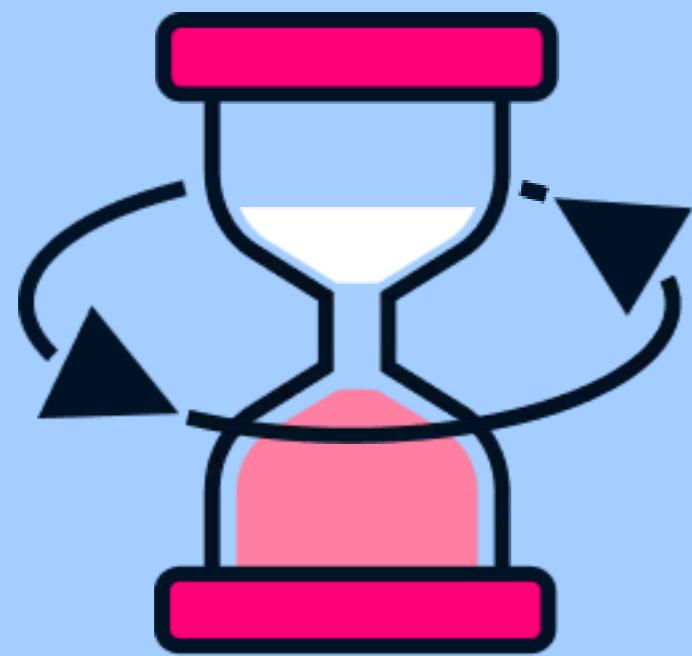


Photo.jpg



My\_Life.txt

# Department Table



```
CREATE TABLE department (
    departmentid INTEGER PRIMARY KEY,
    name VARCHAR(50),
    head INTEGER
);

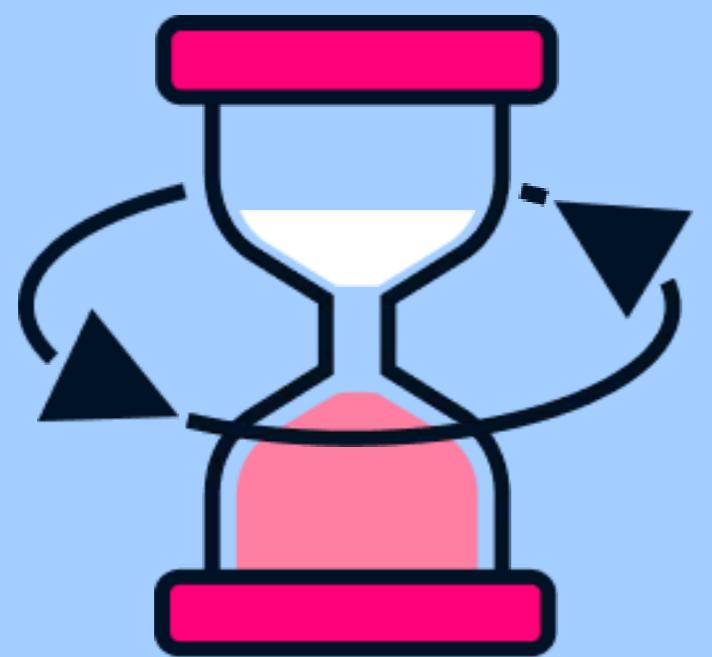
-- Department table
INSERT INTO department (departmentid, name, head) VALUES (1, 'Cardiology', 1);
INSERT INTO department (departmentid, name, head) VALUES (2, 'Neurology', 2);
INSERT INTO department (departmentid, name, head) VALUES (3, 'Oncology', 3);
INSERT INTO department (departmentid, name, head) VALUES (4, 'Pediatrics', 4);
INSERT INTO department (departmentid, name, head) VALUES (5, 'Orthopedics', 5);
INSERT INTO department (departmentid, name, head) VALUES (6, 'Gastroenterology', 6);
INSERT INTO department (departmentid, name, head) VALUES (7, 'Pulmonology', 7);
INSERT INTO department (departmentid, name, head) VALUES (8, 'Endocrinology', 8);
INSERT INTO department (departmentid, name, head) VALUES (9, 'Radiology', 9);
INSERT INTO department (departmentid, name, head) VALUES (10, 'Dermatology', 10);
INSERT INTO department (departmentid, name, head) VALUES (11, 'Urology', 11);
INSERT INTO department (departmentid, name, head) VALUES (12, 'Ophthalmology', 12);
INSERT INTO department (departmentid, name, head) VALUES (13, 'ENT', 13);
INSERT INTO department (departmentid, name, head) VALUES (14, 'Psychiatry', 14);
INSERT INTO department (departmentid, name, head) VALUES (15, 'Anesthesiology', 15);
INSERT INTO department (departmentid, name, head) VALUES (16, 'Pathology', 16);
INSERT INTO department (departmentid, name, head) VALUES (17, 'Emergency Medicine', 17);
INSERT INTO department (departmentid, name, head) VALUES (18, 'Family Medicine', 18);
INSERT INTO department (departmentid, name, head) VALUES (19, 'Internal Medicine', 19);
INSERT INTO department (departmentid, name, head) VALUES (20, 'General Surgery', 20);
```

# Output



departmentid	name	head
1	Cardiology	1
2	Neurology	2
3	Oncology	3
4	Pediatrics	4
5	Orthopedics	5
6	Gastroenterology	6
7	Pulmonology	7
8	Endocrinology	8
9	Radiology	9
1	Dermatology	10
11	Urology	11
12	Ophthalmology	12
13	ENT	13
14	Psychiatry	14
15	Anesthesiology	15
16	Pathology	16
17	Emergency Medicine	17
18	Family Medicine	18
19	Internal Medicine	19
20	General Surgery	20

# Physication Table

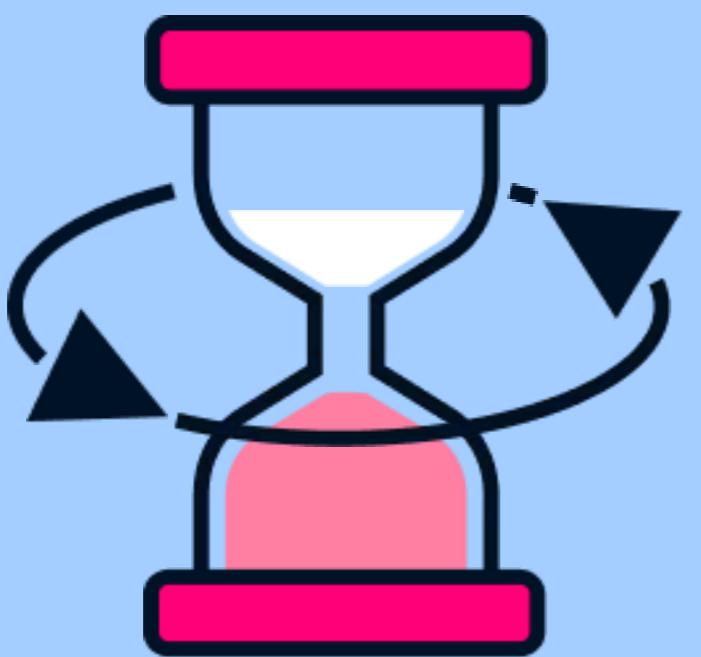


```
CREATE TABLE physician (
    employeeid INTEGER PRIMARY KEY,
    name VARCHAR(50),
    position VARCHAR(50),
    ssn INTEGER
);
-- Physician table
INSERT INTO physician (employeeid, name, position, ssn) VALUES (1, 'Dr. John Smith', 'Cardiologist', 111111111);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (2, 'Dr. Jane Doe', 'Neurologist', 222222222);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (3, 'Dr. Robert Johnson', 'Oncologist', 333333333);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (4, 'Dr. Emily Williams', 'Pediatrician', 444444444);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (5, 'Dr. Michael Brown', 'Orthopedic Surgeon', 555555555);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (6, 'Dr. Sarah Davis', 'Gastroenterologist', 666666666);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (7, 'Dr. David Miller', 'Pulmonologist', 777777777);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (8, 'Dr. Jennifer Wilson', 'Endocrinologist', 888888888);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (9, 'Dr. Richard Moore', 'Radiologist', 999999999);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (10, 'Dr. Lisa Taylor', 'Dermatologist', 101010101);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (11, 'Dr. Thomas Anderson', 'Urologist', 121212121);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (12, 'Dr. Nancy Thomas', 'Ophthalmologist', 131313131);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (13, 'Dr. Daniel Jackson', 'ENT Specialist', 141414141);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (14, 'Dr. Karen White', 'Psychiatrist', 151515151);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (15, 'Dr. Christopher Harris', 'Anesthesiologist', 161616161);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (16, 'Dr. Amanda Martin', 'Pathologist', 171717171);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (17, 'Dr. Matthew Thompson', 'Emergency Physician', 181818181);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (18, 'Dr. Stephanie Garcia', 'Family Physician', 191919191);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (19, 'Dr. Kevin Martinez', 'Internist', 202020202);
INSERT INTO physician (employeeid, name, position, ssn) VALUES (20, 'Dr. Jessica Robinson', 'General Surgeon', 212121212);
```

# Output

employeeid	name	position	ssn
1	Dr. John Smith	Cardiologist	1111111111
2	Dr. Jane Doe	Neurologist	2222222222
3	Dr. Robert Johnson	Oncologist	3333333333
4	Dr. Emily Williams	Pediatrician	4444444444
5	Dr. Michael Brown	Orthopedic Surgeon	5555555555
6	Dr. Sarah Davis	Gastroenterologist	6666666666
7	Dr. David Miller	Pulmonologist	7777777777
8	Dr. Jennifer Wilson	Endocrinologist	8888888888
9	Dr. Richard Moore	Radiologist	9999999999
10	Dr. Lisa Taylor	Dermatologist	101010101
11	Dr. Thomas Anderson	Urologist	121212121
12	Dr. Nancy Thomas	Ophthalmologist	131313131
13	Dr. Daniel Jackson	ENT Specialist	141414141
14	Dr. Karen White	Psychiatrist	151515151
15	Dr. Christopher Harris	Anesthesiologist	161616161
16	Dr. Amanda Martin	Pathologist	171717171
17	Dr. Matthew Thompson	Emergency Physician	181818181
18	Dr. Stephanie Garcia	Family Physician	191919191
19	Dr. Kevin Martinez	Internist	202020202
20	Dr. Jessica Robinson	General Surgeon	212121212

# Nurse Table

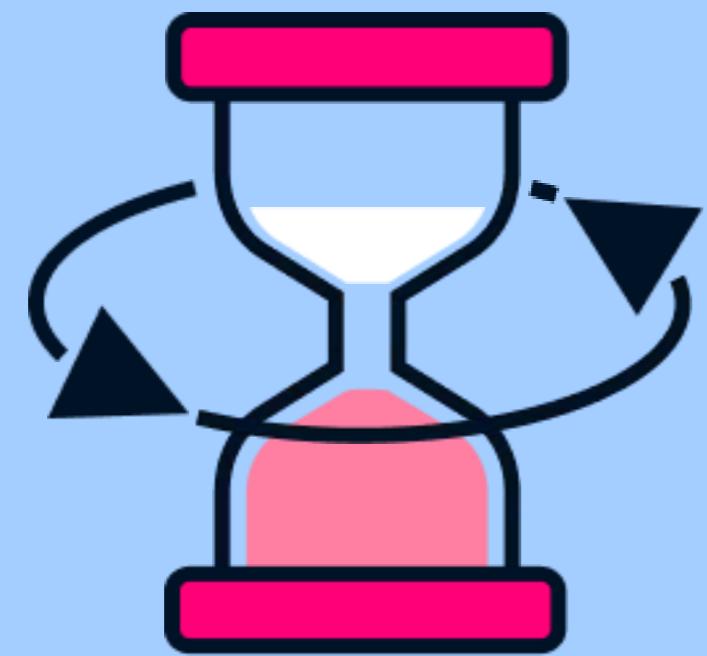


```
CREATE TABLE nurse (
    employeeid INTEGER PRIMARY KEY,
    name VARCHAR(50),
    position VARCHAR(50),
    registered BOOLEAN,
    ssn INTEGER
);
-- Nurse table
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (1, 'Nurse Amy Pond', 'RN', 1, 333333333);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (2, 'Nurse Rory Williams', 'LPN', 0, 444444444);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (3, 'Nurse Clara Oswald', 'RN', 1, 555555555);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (4, 'Nurse Martha Jones', 'LPN', 0, 666666666);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (5, 'Nurse Donna Noble', 'RN', 1, 777777777);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (6, 'Nurse Rose Tyler', 'RN', 1, 888888888);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (7, 'Nurse River Song', 'NP', 1, 999999999);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (8, 'Nurse Jackie Tyler', 'LPN', 0, 101010101);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (9, 'Nurse Sarah Jane Smith', 'RN', 1, 121212121);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (10, 'Nurse Jo Grant', 'LPN', 0, 131313131);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (11, 'Nurse Liz Shaw', 'RN', 1, 141414141);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (12, 'Nurse Leela', 'LPN', 0, 151515151);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (13, 'Nurse Romana', 'RN', 1, 161616161);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (14, 'Nurse Nyssa', 'NP', 1, 171717171);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (15, 'Nurse Tegan Jovanka', 'RN', 1, 181818181);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (16, 'Nurse Peri Brown', 'LPN', 0, 191919191);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (17, 'Nurse Mel Bush', 'RN', 1, 202020202);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (18, 'Nurse Ace McShane', 'LPN', 0, 212121212);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (19, 'Nurse Grace Holloway', 'RN', 1, 232323232);
INSERT INTO nurse (employeeid, name, position, registered, ssn) VALUES (20, 'Nurse Chang Lee', 'LPN', 0, 242424242);
```

# Output

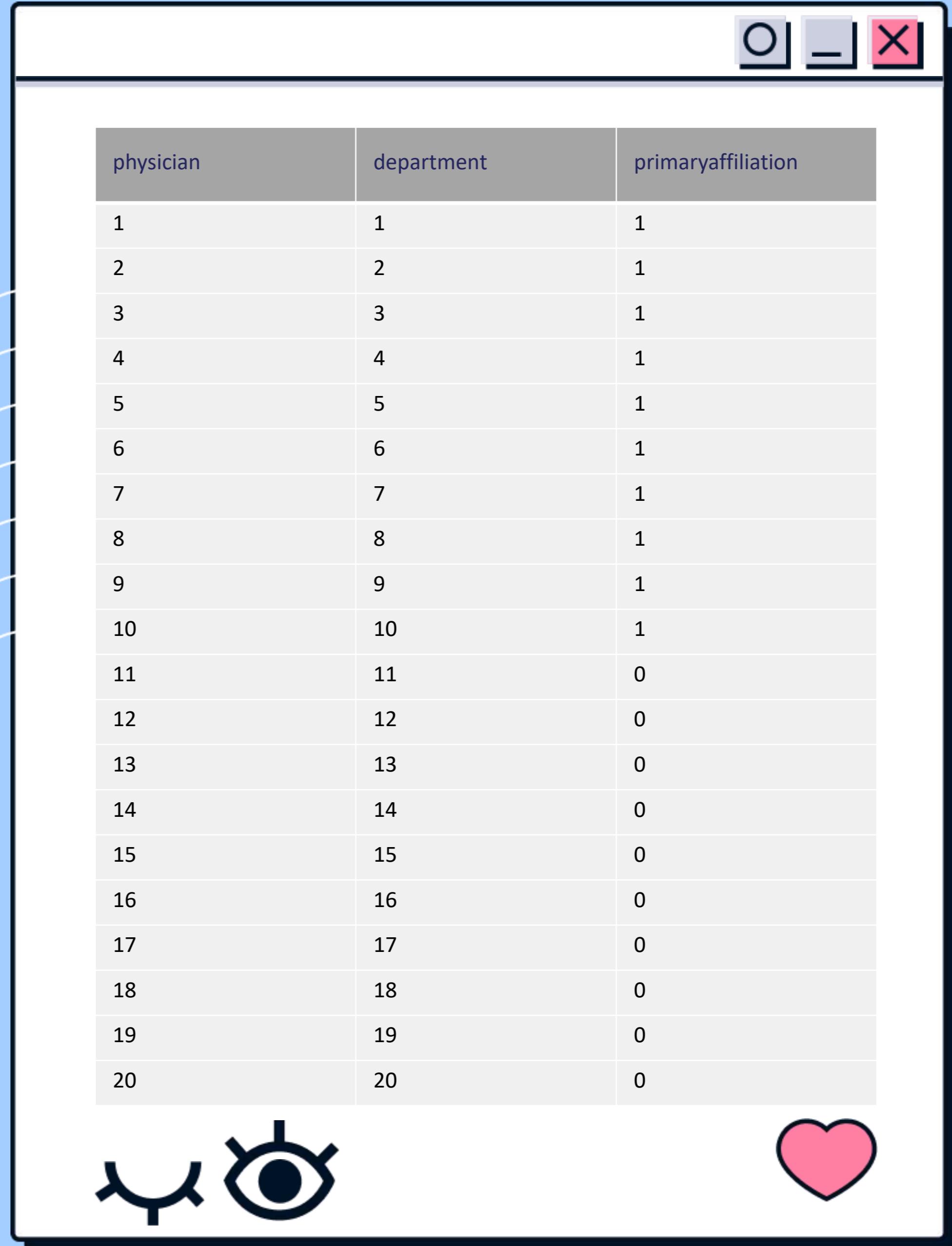
employeeid	name	position	registered	ssn
1	Nurse Amy Pond	RN	1	333333333
2	Nurse Rory Williams	LPN	0	444444444
3	Nurse Clara Oswald	RN	1	555555555
4	Nurse Martha Jones	LPN	0	666666666
5	Nurse Donna Noble	RN	1	777777777
6	Nurse Rose Tyler	RN	1	888888888
7	Nurse River Song	NP	1	999999999
8	Nurse Jackie Tyler	LPN	0	101010101
9	Nurse Sarah Jane Smith	RN	1	121212121
10	Nurse Jo Grant	LPN	0	131313131
11	Nurse Liz Shaw	RN	1	141414141
12	Nurse Leela	LPN	0	151515151
13	Nurse Romana	RN	1	161616161
14	Nurse Nyssa	NP	1	171717171
15	Nurse Tegan Jovanka	RN	1	181818181
16	Nurse Peri Brown	LPN	0	191919191
17	Nurse Mel Bush	RN	1	202020202
18	Nurse Ace McShane	LPN	0	212121212
19	Nurse Grace Holloway	RN	1	232323232
20	Nurse Chang Lee	LPN	0	242424242

# Affiliated\_with Table



```
CREATE TABLE affiliated_with (
    physician INTEGER,
    department INTEGER,
    primaryaffiliation BOOLEAN,
    PRIMARY KEY (physician, department),
    FOREIGN KEY (physician) REFERENCES physician(employeeid),
    FOREIGN KEY (department) REFERENCES department(departmentid)
);
--Records for affiliated_with
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (1, 1, 1);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (2, 2, 1);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (3, 3, 1);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (4, 4, 1);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (5, 5, 1);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (6, 6, 1);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (7, 7, 1);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (10, 10, 1);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (11, 11, 0);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (12, 12, 0);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (13, 13, 0);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (14, 14, 0);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (15, 15, 0);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (16, 16, 0);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (17, 17, 0);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (18, 18, 0);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (19, 19, 0);
INSERT INTO affiliated_with (physician, department, primaryaffiliation) VALUES (20, 20, 0);
```

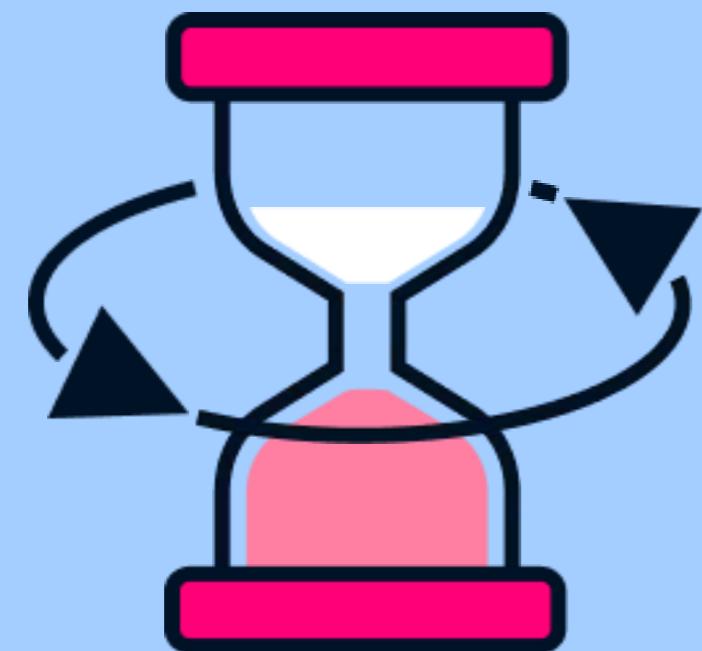
# Output



physician	department	primaryaffiliation
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1
10	10	1
11	11	0
12	12	0
13	13	0
14	14	0
15	15	0
16	16	0
17	17	0
18	18	0
19	19	0
20	20	0



# Block Table



```
CREATE TABLE block (
    blockfloor INTEGER,
    blockcode INTEGER,
    PRIMARY KEY (blockfloor, blockcode)
);
--Records for block
INSERT INTO block (blockfloor, blockcode) VALUES (1, 101);
INSERT INTO block (blockfloor, blockcode) VALUES (1, 102);
INSERT INTO block (blockfloor, blockcode) VALUES (1, 103);
INSERT INTO block (blockfloor, blockcode) VALUES (2, 201);
INSERT INTO block (blockfloor, blockcode) VALUES (2, 202);
INSERT INTO block (blockfloor, blockcode) VALUES (2, 203);
INSERT INTO block (blockfloor, blockcode) VALUES (3, 301);
INSERT INTO block (blockfloor, blockcode) VALUES (3, 302);
INSERT INTO block (blockfloor, blockcode) VALUES (3, 303);
INSERT INTO block (blockfloor, blockcode) VALUES (4, 401);
INSERT INTO block (blockfloor, blockcode) VALUES (4, 402);
INSERT INTO block (blockfloor, blockcode) VALUES (4, 403);
INSERT INTO block (blockfloor, blockcode) VALUES (5, 501);
INSERT INTO block (blockfloor, blockcode) VALUES (5, 502);
INSERT INTO block (blockfloor, blockcode) VALUES (5, 503);
INSERT INTO block (blockfloor, blockcode) VALUES (6, 601);
INSERT INTO block (blockfloor, blockcode) VALUES (6, 602);
INSERT INTO block (blockfloor, blockcode) VALUES (6, 603);
INSERT INTO block (blockfloor, blockcode) VALUES (7, 701);
INSERT INTO block (blockfloor, blockcode) VALUES (7, 702);
```

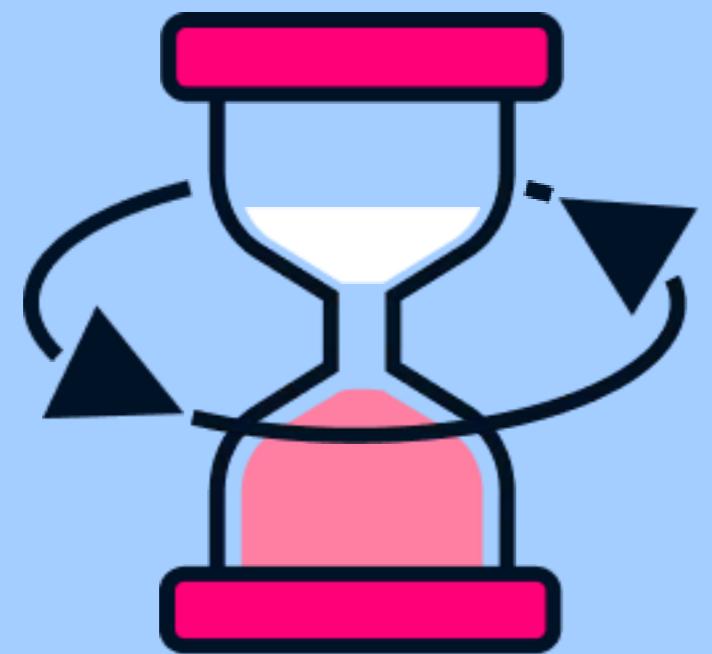


# Output

A screenshot of a software application window. The title bar features three icons: a blue circle with a white outline, a grey square with a white minus sign, and a red square with a white X. The main content is a table with two columns: 'blockfloor' and 'blockcode'. The 'blockfloor' column lists values from 1 to 7. The 'blockcode' column lists corresponding codes: 101, 102, 103, 201, 202, 203, 301, 302, 303, 401, 402, 403, 501, 502, 503, 601, 602, 603, 701, and 702. At the bottom of the window are three icons: a black eye with a curved line, a pink heart, and a small blue icon.

blockfloor	blockcode
1	101
1	102
1	103
2	201
2	202
2	203
3	301
3	302
3	303
4	401
4	402
4	403
5	501
5	502
5	503
6	601
6	602
6	603
7	701
7	702

# Room Table

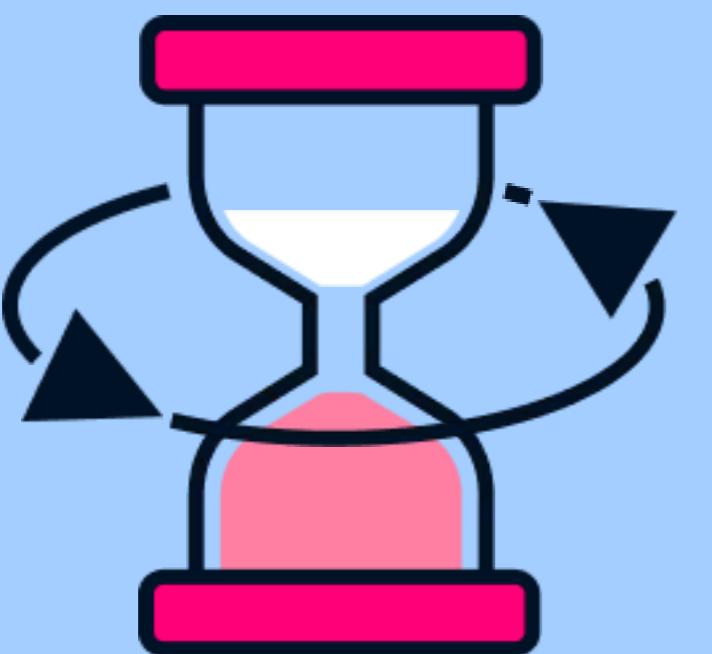


```
CREATE TABLE room (
    roomnumber INTEGER PRIMARY KEY,
    roomtype VARCHAR(50),
    blockfloor INTEGER,
    blockcode INTEGER,
    unavailable BOOLEAN,
    FOREIGN KEY (blockfloor, blockcode) REFERENCES block(blockfloor, blockcode)
);
--Records for room
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (101, 'Single', 1, 101, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (102, 'Single', 1, 101, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (103, 'Double', 1, 102, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (104, 'Double', 1, 102, 1);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (201, 'Single', 2, 201, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (202, 'Single', 2, 201, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (203, 'Double', 2, 202, 1);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (204, 'Double', 2, 202, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (301, 'Single', 3, 301, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (302, 'Single', 3, 301, 1);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (303, 'Double', 3, 302, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (304, 'Double', 3, 302, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (401, 'Single', 4, 401, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (402, 'Single', 4, 401, 1);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (403, 'Double', 4, 402, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (404, 'Double', 4, 402, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (501, 'ICU', 5, 501, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (502, 'ICU', 5, 501, 0);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (503, 'ICU', 5, 502, 1);
INSERT INTO room (roomnumber, roomtype, blockfloor, blockcode, unavailable) VALUES (504, 'ICU', 5, 502, 0);
```

# Output

roomnumber	roomtype	blockfloor	blockcode	unavailable
101	Single	1	101	0
102	Single	1	101	0
103	Double	1	102	0
104	Double	1	102	1
201	Single	2	201	0
202	Single	2	201	0
203	Double	2	202	1
204	Double	2	202	0
301	Single	3	301	0
302	Single	3	301	1
303	Double	3	302	0
304	Double	3	302	0
401	Single	4	401	0
402	Single	4	401	1
403	Double	4	402	0
404	Double	4	402	0
501	ICU	5	501	0
502	ICU	5	501	0
503	ICU	5	502	1
504	ICU	5	502	0

# Patient Table



```
CREATE TABLE patient (
    ssn INTEGER PRIMARY KEY,
    name VARCHAR(50),
    address VARCHAR(200),
    phone VARCHAR(15),
    insuranceid INTEGER,
    pcp INTEGER,
    FOREIGN KEY (pcp) REFERENCES physician(employeeid)
);
--Records for patient
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000001, 'John Smith', '42 Fern St, Boston MA', '555-0101', 68476213, 1);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000002, 'Grace Wong', '12 Maple Ave, Cambridge MA', '555-0102', 36546321, 2);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000003, 'Robert Johnson', '85 Pine Rd, Somerville MA', '555-0103', 65465421, 3);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000004, 'Maria Garcia', '32 Oak St, Medford MA', '555-0104', 68413654, 4);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000005, 'James Wilson', '16 Birch Ln, Malden MA', '555-0105', 54316847, 5);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000006, 'Patricia Lee', '28 Cedar Blvd, Revere MA', '555-0106', 87496546, 6);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000007, 'Michael Brown', '74 Willow Dr, Everett MA', '555-0107', 36498657, 7);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000008, 'Linda Martinez', '55 Spruce Ct, Chelsea MA', '555-0108', 65432187, 8);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000011, 'David Rodriguez', '37 Ash Rd, Milton MA', '555-0111', 13216549, 11);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000012, 'Jennifer Kim', '24 Poplar Ave, Dorchester MA', '555-0112', 65432154, 12);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000013, 'Richard Moore', '18 Beech St, Roxbury MA', '555-0113', 49873216, 13);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000014, 'Susan Jackson', '29 Locust Blvd, Jamaica Plain MA', '555-0114', 21654987, 14);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000015, 'Joseph White', '41 Walnut Dr, Roslindale MA', '555-0115', 65789432, 15);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000016, 'Margaret Harris', '67 Sycamore Ct, Hyde Park MA', '555-0116', 32165498, 16);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000017, 'Thomas Martin', '53 Cypress Rd, West Roxbury MA', '555-0117', 98746513, 17);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000018, 'Dorothy Thompson', '82 Juniper Ln, Brighton MA', '555-0118', 65432198, 18);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000019, 'Charles Robinson', '47 Hemlock St, Allston MA', '555-0119', 32165487, 19);
INSERT INTO patient (ssn, name, address, phone, insuranceid, pcp) VALUES (10000020, 'Barbara Clark', '15 Linden Ave, Brookline MA', '555-0120', 98765431, 20);
```

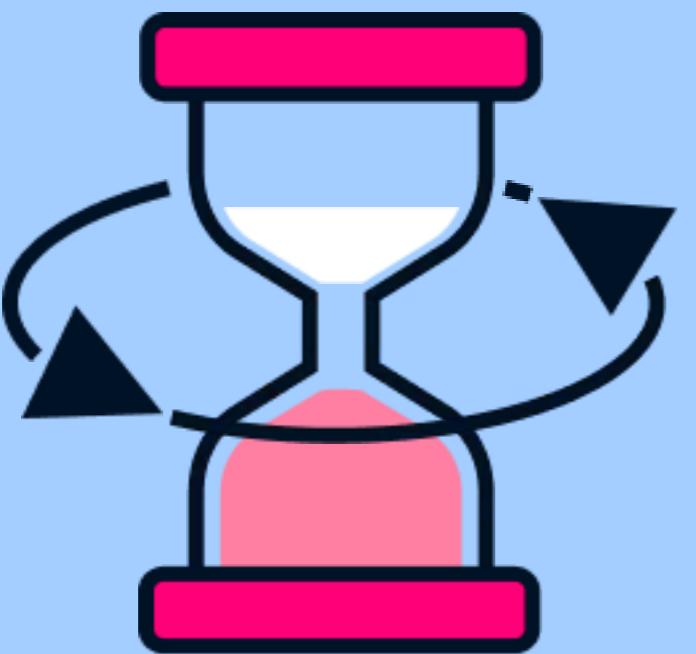
# Output



ssn	name	address	phone	insuranceid	pcp
100000001	John Smith	42 Fern St, Boston MA	555-0101	68476213	1
100000002	Grace Wong	12 Maple Ave, Cambridge MA	555-0102	36546321	2
100000003	Robert Johnson	85 Pine Rd, Somerville MA	555-0103	65465421	3
100000004	Maria Garcia	32 Oak St, Medford MA	555-0104	68413654	4
100000005	James Wilson	16 Birch Ln, Malden MA	555-0105	54316847	5
100000006	Patricia Lee	28 Cedar Blvd, Revere MA	555-0106	87496546	6
100000007	Michael Brown	74 Willow Dr, Everett MA	555-0107	36498657	7
100000008	Linda Martinez	55 Spruce Ct, Chelsea MA	555-0108	65432187	8
100000009	William Taylor	63 Elm St, Winthrop MA	555-0109	32168749	9
100000010	Elizabeth Davis	91 Cherry Ln, Quincy MA	555-0110	65487321	10
100000011	David Rodriguez	37 Ash Rd, Milton MA	555-0111	13216549	11
100000012	Jennifer Kim	24 Poplar Ave, Dorchester MA	555-0112	65432154	12
100000013	Richard Moore	18 Beech St, Roxbury MA	555-0113	49873216	13
100000014	Susan Jackson	29 Locust Blvd, Jamaica Plain MA	555-0114	21654987	14
100000015	Joseph White	41 Walnut Dr, Roslindale MA	555-0115	65789432	15
100000016	Margaret Harris	67 Sycamore Ct, Hyde Park MA	555-0116	32165498	16
100000017	Thomas Martin	53 Cypress Rd, West Roxbury MA	555-0117	98746513	17
100000018	Dorothy Thompson	82 Juniper Ln, Brighton MA	555-0118	65432198	18
100000019	Charles Robinson	47 Hemlock St, Allston MA	555-0119	32165487	19
100000020	Barbara Clark	15 Linden Ave, Brookline MA	555-0120	98765431	20



# Medication Table



```
CREATE TABLE medication (
    code INTEGER PRIMARY KEY,
    name VARCHAR(50),
    brand VARCHAR(50),
    description VARCHAR(200)
);
--Records for medication
INSERT INTO medication (code, name, brand, description) VALUES (1, 'Lisinopril', 'Prinivil', 'ACE inhibitor for hypertension');
INSERT INTO medication (code, name, brand, description) VALUES (2, 'Metformin', 'Glucophage', 'Oral diabetes medication');
INSERT INTO medication (code, name, brand, description) VALUES (3, 'Atorvastatin', 'Lipitor', 'Statin medication for cholesterol control');
INSERT INTO medication (code, name, brand, description) VALUES (4, 'Levothyroxine', 'Synthroid', 'Thyroid hormone replacement');
INSERT INTO medication (code, name, brand, description) VALUES (5, 'Amlodipine', 'Norvasc', 'Calcium channel blocker for hypertension');
INSERT INTO medication (code, name, brand, description) VALUES (6, 'Metoprolol', 'Lopressor', 'Beta blocker for hypertension and angina');
INSERT INTO medication (code, name, brand, description) VALUES (7, 'Omeprazole', 'Prilosec', 'Proton pump inhibitor for acid reflux');
INSERT INTO medication (code, name, brand, description) VALUES (8, 'Losartan', 'Cozaar', 'Angiotensin II receptor blocker for hypertension');
INSERT INTO medication (code, name, brand, description) VALUES (9, 'Albuterol', 'Proventil', 'Bronchodilator for asthma');
INSERT INTO medication (code, name, brand, description) VALUES (10, 'Gabapentin', 'Neurontin', 'Anticonvulsant for neuropathic pain');
INSERT INTO medication (code, name, brand, description) VALUES (11, 'Hydrochlorothiazide', 'Microzide', 'Diuretic for hypertension');
INSERT INTO medication (code, name, brand, description) VALUES (12, 'Sertraline', 'Zoloft', 'SSRI antidepressant');
INSERT INTO medication (code, name, brand, description) VALUES (13, 'Amoxicillin', 'Amoxil', 'Antibiotic for bacterial infections');
INSERT INTO medication (code, name, brand, description) VALUES (14, 'Prednisone', 'Deltasone', 'Corticosteroid anti-inflammatory');
INSERT INTO medication (code, name, brand, description) VALUES (15, 'Fluticasone', 'Flonase', 'Corticosteroid nasal spray for allergies');
INSERT INTO medication (code, name, brand, description) VALUES (16, 'Furosemide', 'Lasix', 'Loop diuretic for edema and hypertension');
INSERT INTO medication (code, name, brand, description) VALUES (17, 'Insulin Glargine', 'Lantus', 'Long-acting insulin for diabetes');
INSERT INTO medication (code, name, brand, description) VALUES (18, 'Warfarin', 'Coumadin', 'Anticoagulant to prevent blood clots');
INSERT INTO medication (code, name, brand, description) VALUES (19, 'Citalopram', 'Celexa', 'SSRI antidepressant');
INSERT INTO medication (code, name, brand, description) VALUES (20, 'Azithromycin', 'Zithromax', 'Macrolide antibiotic');
```

# Output



The image shows a digital interface with a table of 20 medications, a top navigation bar with icons, and three icons at the bottom.

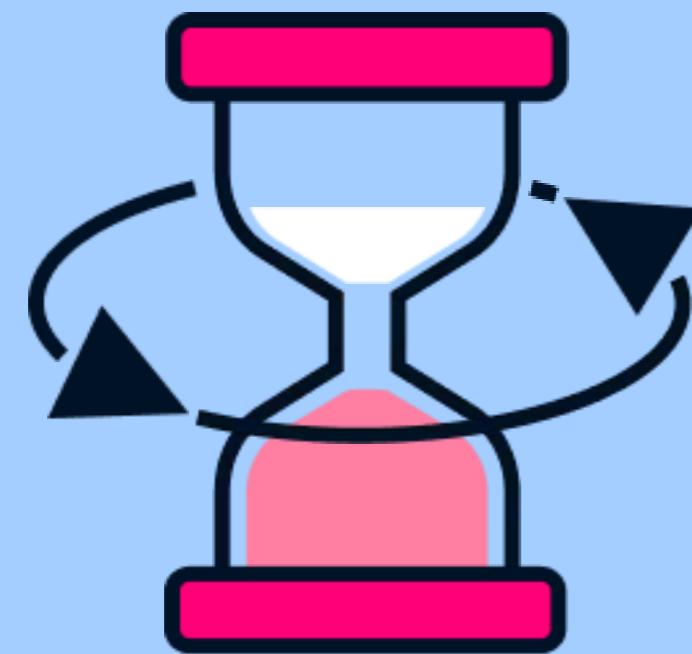
**Top Navigation Bar:** The top right corner features a navigation bar with three icons: a blue circle with a minus sign, a grey square with a minus sign, and a red square with a white X.

**Table:** The main content is a table with 20 rows, each representing a medication. The columns are labeled: code, name, brand, and description.

code	name	brand	description
1	Lisinopril	Prinivil	ACE inhibitor for hypertension
2	Metformin	Glucophage	Oral diabetes medication
3	Atorvastatin	Lipitor	Statin medication for cholesterol control
4	Levothyroxine	Synthroid	Thyroid hormone replacement
5	Amlodipine	Norvasc	Calcium channel blocker for hypertension
6	Metoprolol	Lopressor	Beta blocker for hypertension and angina
7	Omeprazole	Prilosec	Proton pump inhibitor for acid reflux
8	Losartan	Cozaar	Angiotensin II receptor blocker for hypertension
9	Albuterol	Proventil	Bronchodilator for asthma
10	Gabapentin	Neurontin	Anticonvulsant for neuropathic pain
11	Hydrochlorothiazide	Microzide	Diuretic for hypertension
12	Sertraline	Zoloft	SSRI antidepressant
13	Amoxicillin	Amoxil	Antibiotic for bacterial infections
14	Prednisone	Deltasone	Corticosteroid anti-inflammatory
15	Fluticasone	Flonase	Corticosteroid nasal spray for allergies
16	Furosemide	Lasix	Loop diuretic for edema and hypertension
17	Insulin Glargine	Lantus	Long-acting insulin for diabetes
18	Warfarin	Coumadin	Anticoagulant to prevent blood clots
19	Citalopram	Celexa	SSRI antidepressant
20	Azithromycin	Zithromax	Macrolide antibiotic

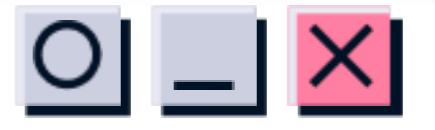
**Bottom Icons:** At the bottom of the screen are three icons: a blue eye-like shape, a black eye-like shape, and a pink heart.

# Procedure Table



```
CREATE TABLE procedure (
    code INTEGER PRIMARY KEY,
    name VARCHAR(50),
    cost REAL
);
--Records for procedure
INSERT INTO procedure (code, name, cost) VALUES (1, 'Appendectomy', 7500.00);
INSERT INTO procedure (code, name, cost) VALUES (2, 'Coronary Bypass', 85000.00);
INSERT INTO procedure (code, name, cost) VALUES (3, 'CT Scan', 1200.00);
INSERT INTO procedure (code, name, cost) VALUES (4, 'MRI', 1500.00);
INSERT INTO procedure (code, name, cost) VALUES (5, 'X-Ray', 350.00);
INSERT INTO procedure (code, name, cost) VALUES (6, 'Blood Test', 75.00);
INSERT INTO procedure (code, name, cost) VALUES (7, 'Colonoscopy', 2200.00);
INSERT INTO procedure (code, name, cost) VALUES (8, 'Endoscopy', 1800.00);
INSERT INTO procedure (code, name, cost) VALUES (9, 'Biopsy', 1600.00);
INSERT INTO procedure (code, name, cost) VALUES (10, 'Cataract Surgery', 4200.00);
INSERT INTO procedure (code, name, cost) VALUES (11, 'Tonsillectomy', 3500.00);
INSERT INTO procedure (code, name, cost) VALUES (12, 'Hernia Repair', 5800.00);
INSERT INTO procedure (code, name, cost) VALUES (13, 'Knee Replacement', 24000.00);
INSERT INTO procedure (code, name, cost) VALUES (14, 'Hip Replacement', 28000.00);
INSERT INTO procedure (code, name, cost) VALUES (15, 'Cardiac Catheterization', 9500.00);
INSERT INTO procedure (code, name, cost) VALUES (16, 'Mammogram', 250.00);
INSERT INTO procedure (code, name, cost) VALUES (17, 'Ultrasound', 400.00);
INSERT INTO procedure (code, name, cost) VALUES (18, 'Physical Therapy (session)', 120.00);
INSERT INTO procedure (code, name, cost) VALUES (19, 'Dialysis (session)', 800.00);
INSERT INTO procedure (code, name, cost) VALUES (20, 'Chemotherapy (session)', 1800.00);
```

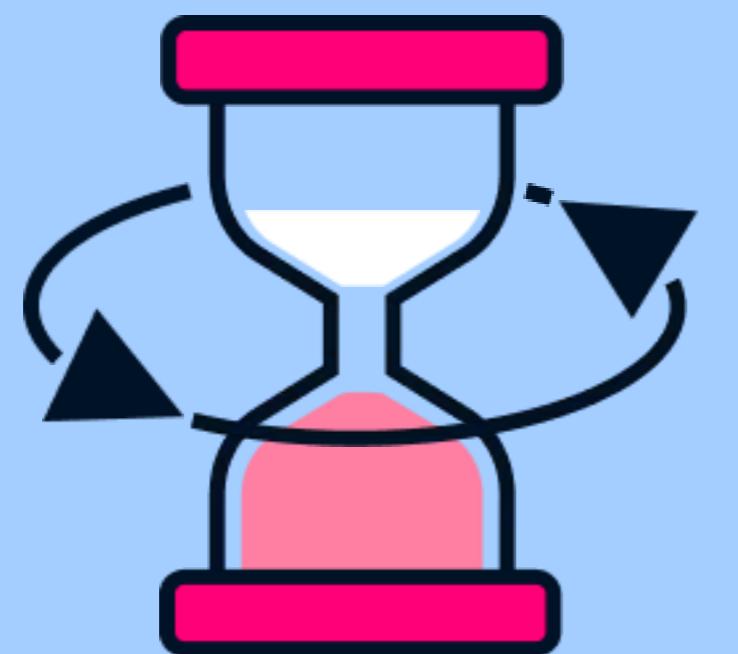
# Output



code	name	cost
1	Appendectomy	7500
2	Coronary Bypass	85000
3	CT Scan	1200
4	MRI	1500
5	X-Ray	350
6	Blood Test	75
7	Colonoscopy	2200
8	Endoscopy	1800
9	Biopsy	1600
10	Cataract Surgery	4200
11	Tonsillectomy	3500
12	Hernia Repair	5800
13	Knee Replacement	24000
14	Hip Replacement	28000
15	Cardiac Catheterization	9500
16	Mammogram	250
17	Ultrasound	400
18	Physical Therapy (session)	120
19	Dialysis (session)	800
20	Chemotherapy (session)	1800



# Trained\_in Table



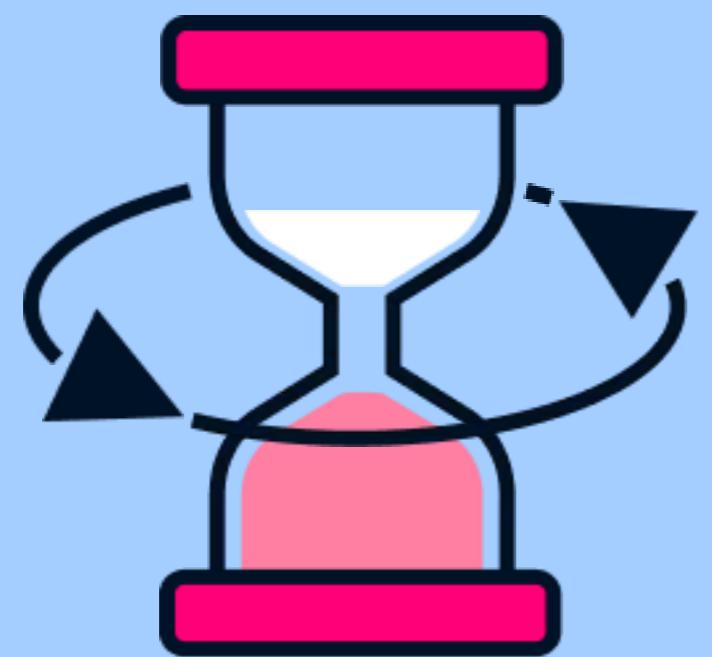
```
CREATE TABLE trained_in (
    physician INTEGER,
    treatment INTEGER,
    certificationdate DATE,
    certificationexpires DATE,
    PRIMARY KEY (physician, treatment),
    FOREIGN KEY (physician) REFERENCES physician(employeeid),
    FOREIGN KEY (treatment) REFERENCES procedure(code)
);

--Records for trained_in
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (1, 1, '2021-05-15', '2026-05-15');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (2, 2, '2020-06-10', '2025-06-10');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (3, 3, '2022-03-22', '2027-03-22');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (4, 4, '2021-09-08', '2026-09-08');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (5, 5, '2023-01-30', '2028-01-30');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (6, 6, '2022-11-12', '2027-11-12');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (7, 7, '2020-08-05', '2025-08-05');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (8, 8, '2021-02-14', '2026-02-14');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (9, 9, '2022-07-23', '2027-07-23');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (10, 10, '2023-04-18', '2028-04-18');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (11, 11, '2020-12-01', '2025-12-01');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (12, 12, '2022-05-19', '2027-05-19');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (13, 13, '2021-10-31', '2026-10-31');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (14, 14, '2023-03-15', '2028-03-15');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (15, 15, '2022-09-22', '2027-09-22');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (16, 16, '2021-04-07', '2026-04-07');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (17, 17, '2023-01-14', '2028-01-14');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (18, 18, '2022-06-28', '2027-06-28');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (19, 19, '2021-11-05', '2026-11-05');
INSERT INTO trained_in (physician, treatment, certificationdate, certificationexpires) VALUES (20, 20, '2023-02-17', '2028-02-17');
```

# Output

physician	treatment	certificationdate	certificationexpires
1	1	2021-05-15	2026-05-15
2	2	2020-06-10	2025-06-10
3	3	2022-03-22	2027-03-22
4	4	2021-09-08	2026-09-08
5	5	2023-01-30	2028-01-30
6	6	2022-11-12	2027-11-12
7	7	2020-08-05	2025-08-05
8	8	2021-02-14	2026-02-14
9	9	2022-07-23	2027-07-23
10	10	2023-04-18	2028-04-18
11	11	2020-12-01	2025-12-01
12	12	2022-05-19	2027-05-19
13	13	2021-10-31	2026-10-31
14	14	2023-03-15	2028-03-15
15	15	2022-09-22	2027-09-22
16	16	2021-04-07	2026-04-07
17	17	2023-01-14	2028-01-14
18	18	2022-06-28	2027-06-28
19	19	2021-11-05	2026-11-05
20	20	2023-02-17	2028-02-17

# Appointment Table

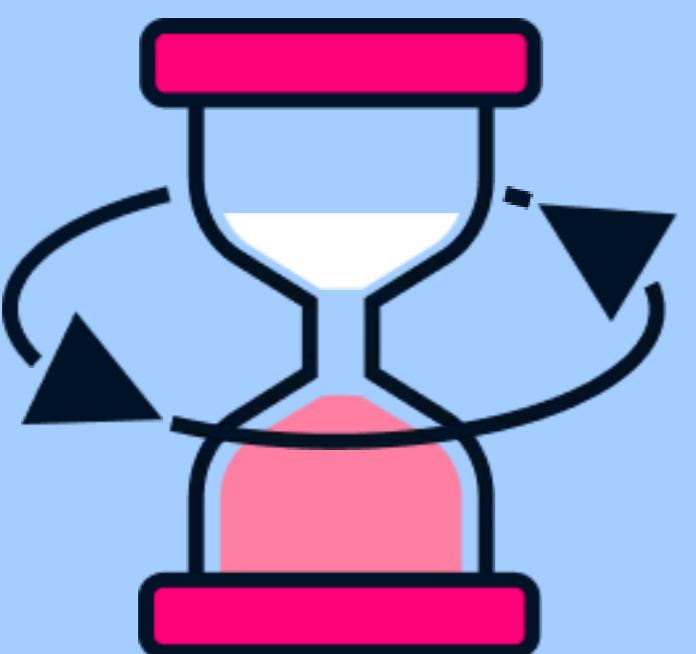


```
CREATE TABLE appointment (
    appointmentid INTEGER PRIMARY KEY,
    patient INTEGER,
    prepuse INTEGER,
    physician INTEGER,
    start_dt_time DATE,
    end_dt_time DATE,
    examinationroom INTEGER,
    FOREIGN KEY (patient) REFERENCES patient(ssn),
    FOREIGN KEY (physician) REFERENCES physician(employeeid)
);
--Records for appointment
INSERT INTO appointment (appointmentid, patient, prepuse, physician, start_dt_time, end_dt_time, examinationroom) VALUES (1, 100000001, 1, 1, '2025-04-25 09:00:00', '2025-04-25 09:30:00', 101);
INSERT INTO appointment (appointmentid, patient, prepuse, physician, start_dt_time, end_dt_time, examinationroom) VALUES (2, 100000002, 2, 2, '2025-04-25 10:00:00', '2025-04-25 10:30:00', 102);
INSERT INTO appointment (appointmentid, patient, prepuse, physician, start_dt_time, end_dt_time, examinationroom) VALUES (3, 100000003, 3, 3, '2025-04-25 11:00:00', '2025-04-25 11:30:00', 103);
INSERT INTO appointment (appointmentid, patient, prepuse, physician, start_dt_time, end_dt_time, examinationroom) VALUES (4, 100000004, 4, 4, '2025-04-25 13:00:00', '2025-04-25 13:30:00', 104);
INSERT INTO appointment (appointmentid, patient, prepuse, physician, start_dt_time, end_dt_time, examinationroom) VALUES (5, 100000005, 5, 5, '2025-04-25 14:00:00', '2025-04-25 14:30:00', 105);
INSERT INTO appointment (appointmentid, patient, prepuse, physician, start_dt_time, end_dt_time, examinationroom) VALUES (6, 100000006, 6, 6, '2025-04-26 09:00:00', '2025-04-26 09:30:00', 101);
INSERT INTO appointment (appointmentid, patient, prepuse, physician, start_dt_time, end_dt_time, examinationroom) VALUES (7, 100000007, 7, 7, '2025-04-26 10:00:00', '2025-04-26 10:30:00', 102);
INSERT INTO appointment (appointmentid, patient, prepuse, physician, start_dt_time, end_dt_time, examinationroom) VALUES (8, 100000008, 8, 8, '2025-04-26 11:00:00', '2025-04-26 11:30:00', 103);
INSERT INTO appointment (appointmentid, patient, prepuse, physician, start_dt_time, end_dt_time, examinationroom) VALUES (9, 100000009, 9, 9, '2025-04-26 13:00:00', '2025-04-26 13:30:00', 104);
INSERT INTO appointment (appointmentid, patient, prepuse, physician, start_dt_time, end_dt_time, examinationroom) VALUES (10, 100000010, 10, 10, '2025-04-26 14:00:00', '2025-04-26 14:30:00', 105);
INSERT INTO appointment (appointmentid, patient, prepuse, physician, start_dt_time, end_dt_time, examinationroom) VALUES (11, 100000011, 11, 11, '2025-04-27 09:00:00', '2025-04-27 09:30:00', 101);
INSERT INTO appointment (appointmentid, patient, prepuse, physician, start_dt_time, end_dt_time, examinationroom) VALUES (12, 100000012, 12, 12, '2025-04-27 10:00:00', '2025-04-27 10:30:00', 102);
```

# Output

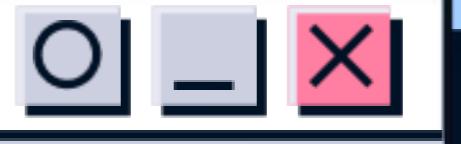
appointmentid	patient	prepuse	physician	start_dt_time	end_dt_time	examinationroom
1	100000001	1	1	2025-04-25 09:00:00	2025-04-25 09:30:00	101
2	100000002	2	2	2025-04-25 10:00:00	2025-04-25 10:30:00	102
3	100000003	3	3	2025-04-25 11:00:00	2025-04-25 11:30:00	103
4	100000004	4	4	2025-04-25 13:00:00	2025-04-25 13:30:00	104
5	100000005	5	5	2025-04-25 14:00:00	2025-04-25 14:30:00	105
6	100000006	6	6	2025-04-26 09:00:00	2025-04-26 09:30:00	101
7	100000007	7	7	2025-04-26 10:00:00	2025-04-26 10:30:00	102
8	100000008	8	8	2025-04-26 11:00:00	2025-04-26 11:30:00	103
9	100000009	9	9	2025-04-26 13:00:00	2025-04-26 13:30:00	104
10	100000010	10	10	2025-04-26 14:00:00	2025-04-26 14:30:00	105
11	100000011	11	11	2025-04-27 09:00:00	2025-04-27 09:30:00	101
12	100000012	12	12	2025-04-27 10:00:00	2025-04-27 10:30:00	102
13	100000013	13	13	2025-04-27 11:00:00	2025-04-27 11:30:00	103
14	100000014	14	14	2025-04-27 13:00:00	2025-04-27 13:30:00	104
15	100000015	15	15	2025-04-27 14:00:00	2025-04-27 14:30:00	105
16	100000016	16	16	2025-04-28 09:00:00	2025-04-28 09:30:00	101
17	100000017	17	17	2025-04-28 10:00:00	2025-04-28 10:30:00	102
18	100000018	18	18	2025-04-28 11:00:00	2025-04-28 11:30:00	103
19	100000019	19	19	2025-04-28 13:00:00	2025-04-28 13:30:00	104
20	100000020	20	20	2025-04-28 14:00:00	2025-04-28 14:30:00	105

# Prescribes Table



```
CREATE TABLE prescribes (
    physician INTEGER,
    patient INTEGER,
    medication INTEGER,
    date DATE,
    appointment INTEGER,
    dose VARCHAR(50),
    PRIMARY KEY (physician, patient, medication, date),
    FOREIGN KEY (physician) REFERENCES physician(employeeid),
    FOREIGN KEY (patient) REFERENCES patient(ssn),
    FOREIGN KEY (medication) REFERENCES medication(code),
    FOREIGN KEY (appointment) REFERENCES appointment(appointmentid)
);
--Records for prescribes
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (1, 10000001, 1, '2025-04-25', 1, '10mg daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (2, 10000002, 2, '2025-04-25', 2, '500mg twice daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (3, 10000003, 3, '2025-04-25', 3, '20mg daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (4, 10000004, 4, '2025-04-25', 4, '75mcg daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (5, 10000005, 5, '2025-04-25', 5, '5mg daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (6, 10000006, 6, '2025-04-26', 6, '50mg twice daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (7, 10000007, 7, '2025-04-26', 7, '20mg daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (8, 10000008, 8, '2025-04-26', 8, '25mg daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (9, 10000009, 9, '2025-04-26', 9, '2 puffs as needed');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (10, 10000010, 10, '2025-04-26', 10, '300mg three times daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (11, 10000011, 11, '2025-04-27', 11, '25mg daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (12, 10000012, 12, '2025-04-27', 12, '50mg');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (13, 10000013, 13, '2025-04-27', 13, '500mg three times daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (14, 10000014, 14, '2025-04-27', 14, '10mg daily for 7 days');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (15, 10000015, 15, '2025-04-27', 15, '1 spray each nostril daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (16, 10000016, 16, '2025-04-28', 16, '40mg daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (17, 10000017, 17, '2025-04-28', 17, '10 units at bedtime');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (18, 10000018, 18, '2025-04-28', 18, '5mg daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (19, 10000019, 19, '2025-04-28', 19, '20mg daily');
INSERT INTO prescribes (physician, patient, medication, date, appointment, dose) VALUES (20, 10000020, 20, '2025-04-28', 20, '500mg on day 1, 250mg for 4 days');
```

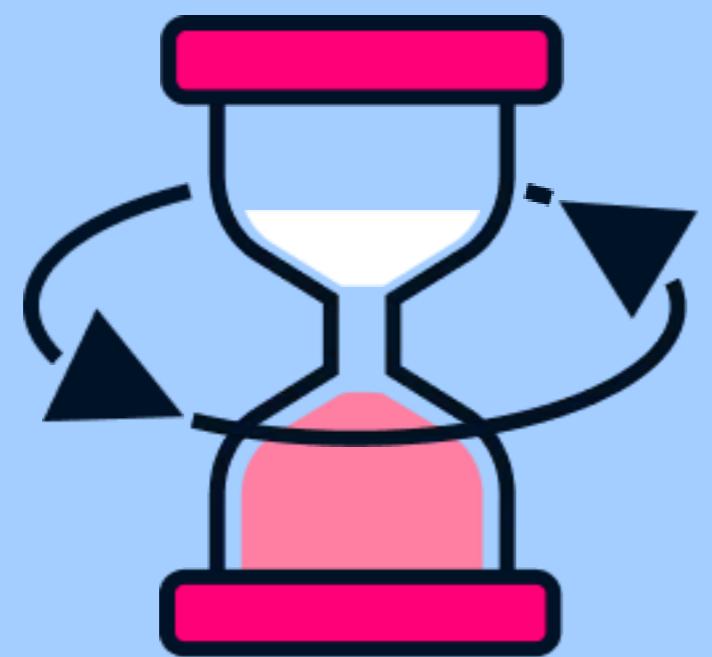
# Output



physician	patient	medication	date	appointment	dose
1	100000001	1	2025-04-25	1	10mg daily
2	100000002	2	2025-04-25	2	500mg twice daily
3	100000003	3	2025-04-25	3	20mg daily
4	100000004	4	2025-04-25	4	75mcg daily
5	100000005	5	2025-04-25	5	5mg daily
6	100000006	6	2025-04-26	6	50mg twice daily
7	100000007	7	2025-04-26	7	20mg daily
8	100000008	8	2025-04-26	8	25mg daily
9	100000009	9	2025-04-26	9	2 puffs as needed
10	100000010	10	2025-04-26	10	300mg three times daily
11	100000011	11	2025-04-27	11	25mg daily
12	100000012	12	2025-04-27	12	50mg daily
13	100000013	13	2025-04-27	13	500mg three times daily
14	100000014	14	2025-04-27	14	10mg daily for 7 days
15	100000015	15	2025-04-27	15	1 spray each nostril daily
16	100000016	16	2025-04-28	16	40mg daily
17	100000017	17	2025-04-28	17	10 units at bedtime
18	100000018	18	2025-04-28	18	5mg daily
19	100000019	19	2025-04-28	19	20mg daily
20	100000020	20	2025-04-28	20	500mg on day 1, 250mg for 4 days



# Stay Table



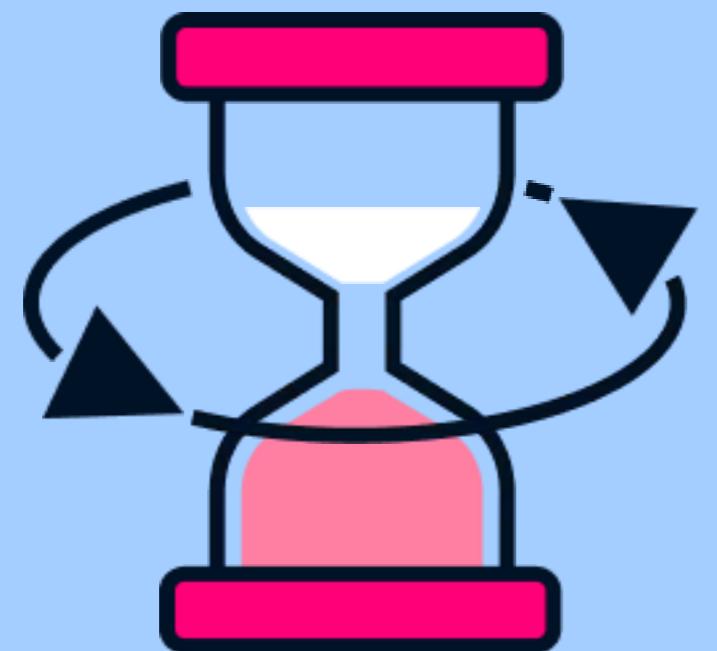
```
CREATE TABLE stay (
    stayid INTEGER PRIMARY KEY,
    patient INTEGER,
    room INTEGER,
    start_time DATE,
    end_time DATE,
    FOREIGN KEY (patient) REFERENCES patient(ssn),
    FOREIGN KEY (room) REFERENCES room(roomnumber)
);
-- Records for stay
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (1, 10000001, 101, '2025-01-01 08:00:00', '2025-01-03 10:30:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (2, 10000002, 102, '2025-01-02 09:15:00', '2025-01-05 14:00:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (3, 10000003, 103, '2025-01-03 11:30:00', '2025-01-04 16:45:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (4, 10000004, 201, '2025-01-03 14:00:00', '2025-01-08 09:00:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (5, 10000005, 202, '2025-01-04 10:00:00', '2025-01-06 11:30:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (6, 10000006, 203, '2025-01-05 08:30:00', '2025-01-07 13:15:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (7, 10000007, 301, '2025-01-06 09:45:00', '2025-01-09 10:00:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (8, 10000008, 302, '2025-01-07 07:30:00', '2025-01-10 08:45:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (12, 10000012, 103, '2025-01-11 16:00:00', '2025-01-14 11:30:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (13, 10000013, 201, '2025-01-12 08:45:00', '2025-01-15 10:45:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (14, 10000014, 202, '2025-01-13 13:00:00', '2025-01-16 14:30:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (15, 10000015, 203, '2025-01-14 10:30:00', '2025-01-17 08:00:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (16, 10000016, 301, '2025-01-15 15:15:00', '2025-01-18 16:45:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (17, 10000017, 302, '2025-01-16 09:30:00', '2025-01-19 10:15:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (18, 10000018, 303, '2025-01-17 07:45:00', '2025-01-20 13:30:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (19, 10000019, 101, '2025-01-18 12:00:00', '2025-01-21 09:00:00');
INSERT INTO stay (stayid, patient, room, start_time, end_time) VALUES (20, 10000020, 102, '2025-01-19 16:30:00', '2025-01-22 11:45:00');
```

# Output



stayid	patient	room	start_time	end_time
1	100000001	101	2025-01-01 08:00:00	2025-01-03 10:30:00
2	100000002	102	2025-01-02 09:15:00	2025-01-05 14:00:00
3	100000003	103	2025-01-03 11:30:00	2025-01-04 16:45:00
4	100000004	201	2025-01-03 14:00:00	2025-01-08 09:00:00
5	100000005	202	2025-01-04 10:00:00	2025-01-06 11:30:00
6	100000006	203	2025-01-05 08:30:00	2025-01-07 13:15:00
7	100000007	301	2025-01-06 09:45:00	2025-01-09 10:00:00
8	100000008	302	2025-01-07 07:30:00	2025-01-10 08:45:00
9	100000009	303	2025-01-08 12:15:00	2025-01-11 15:30:00
10	100000010	101	2025-01-09 11:00:00	2025-01-12 12:00:00
11	100000011	102	2025-01-10 14:30:00	2025-01-13 09:15:00
12	100000012	103	2025-01-11 16:00:00	2025-01-14 11:30:00
13	100000013	201	2025-01-12 08:45:00	2025-01-15 10:45:00
14	100000014	202	2025-01-13 13:00:00	2025-01-16 14:30:00
15	100000015	203	2025-01-14 10:30:00	2025-01-17 08:00:00
16	100000016	301	2025-01-15 15:15:00	2025-01-18 16:45:00
17	100000017	302	2025-01-16 09:30:00	2025-01-19 10:15:00
18	100000018	303	2025-01-17 07:45:00	2025-01-20 13:30:00
19	100000019	101	2025-01-18 12:00:00	2025-01-21 09:00:00
20	100000020	102	2025-01-19 16:30:00	2025-01-22 11:45:00

# On\_Call Table

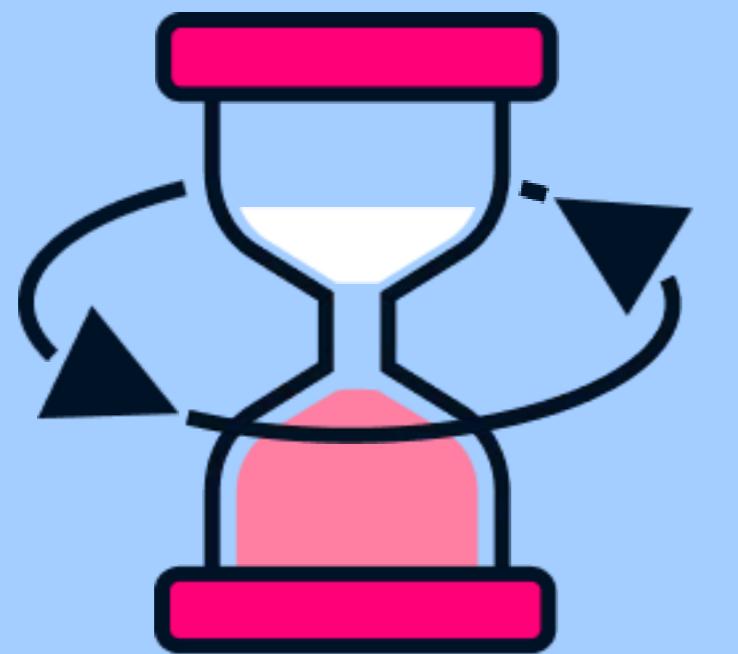


```
CREATE TABLE on_call (
    nurse INTEGER,
    blockfloor INTEGER,
    blockcode INTEGER,
    oncallstart DATE,
    oncallend DATE,
    PRIMARY KEY (nurse, blockfloor, blockcode, oncallstart, oncallend),
    FOREIGN KEY (nurse) REFERENCES nurse(employeeid),
    FOREIGN KEY (blockfloor, blockcode) REFERENCES block(blockfloor, blockcode)
);
-- Records for on_call
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (1, 1, 101, '2025-01-01 08:00:00', '2025-01-01 20:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (2, 2, 102, '2025-01-01 20:00:00', '2025-01-02 08:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (3, 3, 103, '2025-01-02 08:00:00', '2025-01-02 20:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (4, 1, 201, '2025-01-02 20:00:00', '2025-01-03 08:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (5, 2, 202, '2025-01-03 08:00:00', '2025-01-03 20:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (6, 3, 203, '2025-01-03 20:00:00', '2025-01-04 08:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (7, 1, 301, '2025-01-04 08:00:00', '2025-01-04 20:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (8, 2, 302, '2025-01-04 20:00:00', '2025-01-05 08:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (9, 3, 303, '2025-01-05 08:00:00', '2025-01-05 20:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (10, 1, 101, '2025-01-05 20:00:00', '2025-01-06 08:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (1, 2, 102, '2025-01-06 08:00:00', '2025-01-06 20:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (2, 3, 103, '2025-01-06 20:00:00', '2025-01-07 08:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (3, 1, 201, '2025-01-07 08:00:00', '2025-01-07 20:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (4, 2, 202, '2025-01-07 20:00:00', '2025-01-08 08:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (5, 3, 203, '2025-01-08 08:00:00', '2025-01-08 20:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (6, 1, 301, '2025-01-08 20:00:00', '2025-01-09 08:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (7, 2, 302, '2025-01-09 08:00:00', '2025-01-09 20:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (8, 3, 303, '2025-01-09 20:00:00', '2025-01-10 08:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (9, 1, 101, '2025-01-10 08:00:00', '2025-01-10 20:00:00');
INSERT INTO on_call (nurse, blockfloor, blockcode, oncallstart, oncallend) VALUES (10, 2, 102, '2025-01-10 20:00:00', '2025-01-11 08:00:00');
```



# Output

# Undergoes Table



```
CREATE TABLE undergoes (
    patient INTEGER,
    procedure INTEGER,
    stay INTEGER,
    date DATE,
    physician INTEGER,
    assistingnurse INTEGER,
    PRIMARY KEY (patient, procedure, stay, date),
    FOREIGN KEY (patient) REFERENCES patient(ssn),
    FOREIGN KEY (procedure) REFERENCES procedure(code),
    FOREIGN KEY (stay) REFERENCES stay(stayid),
    FOREIGN KEY (physician) REFERENCES physician(employeeid),
    FOREIGN KEY (assistingnurse) REFERENCES nurse(employeeid)
);
-- Records for undergoes
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000001, 1, 1, '2025-01-02', 1, 1);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000002, 2, 2, '2025-01-03', 2, 2);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000003, 3, 3, '2025-01-04', 3, 3);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000004, 4, 4, '2025-01-05', 4, 4);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000005, 5, 5, '2025-01-05', 5, 5);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000006, 1, 6, '2025-01-06', 6, 6);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000007, 2, 7, '2025-01-07', 7, 7);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000008, 3, 8, '2025-01-08', 8, 8);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000009, 4, 9, '2025-01-09', 9, 9);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000010, 5, 10, '2025-01-10', 10, 10);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000011, 1, 11, '2025-01-11', 1, 1);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000012, 2, 12, '2025-01-12', 2, 2);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000013, 3, 13, '2025-01-13', 3, 3);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000014, 4, 14, '2025-01-14', 4, 4);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000015, 5, 15, '2025-01-15', 5, 5);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000016, 1, 16, '2025-01-16', 6, 6);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000017, 2, 17, '2025-01-17', 7, 7);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000018, 3, 18, '2025-01-18', 8, 8);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000019, 4, 19, '2025-01-19', 9, 9);
INSERT INTO undergoes (patient, procedure, stay, date, physician, assistingnurse) VALUES (100000020, 5, 20, '2025-01-20', 10, 10);
```

# Output



patient	procedure	stay	date	physician	assistingnurse
100000001	1	1	2025-01-02	1	1
100000002	2	2	2025-01-03	2	2
100000003	3	3	2025-01-04	3	3
100000004	4	4	2025-01-05	4	4
100000005	5	5	2025-01-05	5	5
100000006	1	6	2025-01-06	6	6
100000007	2	7	2025-01-07	7	7
100000008	3	8	2025-01-08	8	8
100000009	4	9	2025-01-09	9	9
100000010	5	10	2025-01-10	10	10
100000011	1	11	2025-01-11	1	1
100000012	2	12	2025-01-12	2	2
100000013	3	13	2025-01-13	3	3
100000014	4	14	2025-01-14	4	4
100000015	5	15	2025-01-15	5	5
100000016	1	16	2025-01-16	6	6
100000017	2	17	2025-01-17	7	7
100000018	3	18	2025-01-18	8	8
100000019	4	19	2025-01-19	9	9
100000020	5	20	2025-01-20	10	10

01

Write a query in SQL to find all the information of the nurses who are yet to be registered.



```
SELECT *  
FROM nurse  
WHERE registered = FALSE;
```

employeeid	name	position	registered	ssn
2	Nurse Rory Williams	LPN	0	444444444
4	Nurse Martha Jones	LPN	0	666666666
8	Nurse Jackie Tyler	LPN	0	101010101
10	Nurse Jo Grant	LPN	0	131313131
12	Nurse Leela	LPN	0	151515151
16	Nurse Peri Brown	LPN	0	191919191
18	Nurse Ace McShane	LPN	0	212121212
20	Nurse Chang Lee	LPN	0	242424242



02

Write a query in SQL to find the name of the nurse who are the head of their department.



```
SELECT n.name AS nurse_name, d.name AS  
department_name  
FROM nurse n  
JOIN department d ON n.employeeid = d.head;
```

nurse_name	department_name
Nurse Amy Pond	Cardiology
Nurse Rory Williams	Neurology
Nurse Clara Oswald	Oncology
Nurse Martha Jones	Pediatrics
Nurse Donna Noble	Orthopedics
Nurse Rose Tyler	Gastroenterology
Nurse River Song	Pulmonology
Nurse Jackie Tyler	Endocrinology
Nurse Sarah Jane Smith	Radiology
Nurse Jo Grant	Dermatology
Nurse Liz Shaw	Urology
Nurse Leela	Ophthalmology
Nurse Romana	ENT
Nurse Nyssa	Psychiatry
Nurse Tegan Jovanka	Anesthesiology
Nurse Peri Brown	Pathology
Nurse Mel Bush	Emergency Medicine

03

Write a query in SQL to obtain the name of the physicians who are the head of each department.



```
SELECT p.name AS physician_name, d.name AS  
department_name  
FROM physician p  
JOIN department d ON p.employeeid = d.head  
ORDER BY d.name;
```

physician_name	department_name
Dr. Christopher Harris	Anesthesiology
Dr. John Smith	Cardiology
Dr. Lisa Taylor	Dermatology
Dr. Daniel Jackson	ENT
Dr. Matthew Thompson	Emergency Medicine
Dr. Jennifer Wilson	Endocrinology
Dr. Stephanie Garcia	Family Medicine
Dr. Sarah Davis	Gastroenterology
Dr. Jessica Robinson	General Surgery
Dr. Kevin Martinez	Internal Medicine
Dr. Jane Doe	Neurology
Dr. Robert Johnson	Oncology
Dr. Nancy Thomas	Ophthalmology
Dr. Michael Brown	Orthopedics
Dr. Amanda Martin	Pathology
Dr. Emily Williams	Pediatrics
Dr. Karen White	Psychiatry



04

Write a query in SQL to count the number of patients who taken appointment with at least one physician.

```
SELECT COUNT(DISTINCT patient) AS  
patients_with_appointments  
  
FROM appointment;
```

: patients\_with\_appointments

20



05

Write a query in SQL to find the floor and block where the room number 202 belongs to.

```
SELECT roomnumber, blockfloor AS floor,  
blockcode AS block  
  
FROM room  
  
WHERE roomnumber = 202;
```

roomnumber	floor	block
202	2	201



06

Write a query in SQL to count the number available rooms

```
SELECT COUNT(*) AS available_rooms  
FROM room  
WHERE unavailable = FALSE;
```

```
: available_rooms  
15
```



07

Write a query in SQL to count the number of unavailable rooms.

```
SELECT COUNT(*) AS available_rooms  
FROM room  
WHERE unavailable = TRUE;
```

available\_rooms



08

Write a query in SQL to obtain the name of the physician and the departments they are affiliated with.



```
SELECT p.name AS physician_name, d.name AS department_name,  
aw.primaryaffiliation AS is_primary  
FROM physician p  
JOIN affiliated_with aw ON p.employeeid = aw.physician  
JOIN department d ON aw.department = d.departmentid  
ORDER BY p.name, d.name;
```

physician_name	department_name	is_primary
Dr. Amanda Martin	Pathology	0
Dr. Christopher Harris	Anesthesiology	0
Dr. Daniel Jackson	ENT	0
Dr. David Miller	Pulmonology	1
Dr. Emily Williams	Pediatrics	1
Dr. Jane Doe	Neurology	1
Dr. Jennifer Wilson	Endocrinology	1
Dr. Jessica Robinson	General Surgery	0
Dr. John Smith	Cardiology	1
Dr. Karen White	Psychiatry	0
Dr. Kevin Martinez	Internal Medicine	0
Dr. Lisa Taylor	Dermatology	1
Dr. Matthew Thompson	Emergency Medicine	0
Dr. Michael Brown	Orthopedics	1
Dr. Nancy Thomas	Ophthalmology	0
Dr. Richard Moore	Radiology	1
Dr. Robert Johnson	Oncology	1



09

Write a query in SQL to obtain the name of the physicians who are trained for a special treatment.



```
SELECT p.name AS physician_name, pr.name AS treatment_name,  
ti.certificationdate, ti.certificationexpires  
FROM physician p  
JOIN trained_in ti ON p.employeeid = ti.physician  
JOIN procedure pr ON ti.treatment = pr.code  
ORDER BY p.name, pr.name;
```

physician_name	treatment_name	certificationdate	certificationexpires
Dr. Amanda Martin	Mammogram	2021-04-07	2026-04-07
Dr. Christopher Harris	Cardiac Catheterization	2022-09-22	2027-09-22
Dr. Daniel Jackson	Knee Replacement	2021-10-31	2026-10-31
Dr. David Miller	Colonoscopy	2020-08-05	2025-08-05
Dr. Emily Williams	MRI	2021-09-08	2026-09-08
Dr. Jane Doe	Coronary Bypass	2020-06-10	2025-06-10
Dr. Jennifer Wilson	Endoscopy	2021-02-14	2026-02-14
Dr. Jessica Robinson	Chemotherapy (session)	2023-02-17	2028-02-17
Dr. John Smith	Appendectomy	2021-05-15	2026-05-15
Dr. Karen White	Hip Replacement	2023-03-15	2028-03-15
Dr. Kevin Martinez	Dialysis (session)	2021-11-05	2026-11-05
Dr. Lisa Taylor	Cataract Surgery	2023-04-18	2028-04-18
Dr. Matthew Thompson	Ultrasound	2023-01-14	2028-01-14
Dr. Michael Brown	X-Ray	2023-01-30	2028-01-30
Dr. Nancy Thomas	Hernia Repair	2022-05-19	2027-05-19
Dr. Richard Moore	Biopsy	2022-07-23	2027-07-23
Dr. Robert Johnson	CT Scan	2022-03-22	2027-03-22

10

Write a query in SQL to obtain the name of the physicians with department who are yet to be affiliated.

```
SELECT p.employeeid, p.name AS physician_name, p.position  
FROM physician p  
LEFT JOIN affiliated_with aw ON p.employeeid = aw.physician  
WHERE aw.physician IS NULL;
```

employeeid	physician_name	position



11

Write a query in SQL to obtain the name of the physicians who are not a specialized physician.

```
SELECT p.employeeid, p.name AS  
physician_name, p.position  
  
FROM physician p  
  
LEFT JOIN trained_in ti ON p.employeeid =  
ti.physician  
  
WHERE ti.physician IS NULL;
```

employeeid	physician_name	position



12

Write a query in SQL to obtain the name of the patients with their physicians by whom they got their preliminary treatment.



```
SELECT pt.name AS patient_name, ph.name AS primary_care_physician, ph.position AS physician_specialty  
FROM patient pt  
JOIN physician ph ON pt.pcp = ph.employeeid  
ORDER BY pt.name;
```

patient_name	primary_care_physician	physician_specialty
Barbara Clark	Dr. Jessica Robinson	General Surgeon
Charles Robinson	Dr. Kevin Martinez	Internist
David Rodriguez	Dr. Thomas Anderson	Urologist
Dorothy Thompson	Dr. Stephanie Garcia	Family Physician
Elizabeth Davis	Dr. Lisa Taylor	Dermatologist
Grace Wong	Dr. Jane Doe	Neurologist
James Wilson	Dr. Michael Brown	Orthopedic Surgeon
Jennifer Kim	Dr. Nancy Thomas	Ophthalmologist
John Smith	Dr. John Smith	Cardiologist
Joseph White	Dr. Christopher Harris	Anesthesiologist
Linda Martinez	Dr. Jennifer Wilson	Endocrinologist
Margaret Harris	Dr. Amanda Martin	Pathologist
Maria Garcia	Dr. Emily Williams	Pediatrician
Michael Brown	Dr. David Miller	Pulmonologist
Patricia Lee	Dr. Sarah Davis	Gastroenterologist
Richard Moore	Dr. Daniel Jackson	ENT Specialist
Robert Johnson	Dr. Robert Johnson	Oncologist

13

Write a query in SQL to find the name of the patients and the number of physicians they have taken appointment.



```
SELECT p.name AS patient_name, COUNT(DISTINCT a.physician) AS  
number_of_physicians  
FROM patient p  
JOIN appointment a ON p.ssn = a.patient  
GROUP BY p.ssn, p.name  
ORDER BY number_of_physicians DESC, p.name;
```

patient_name	number_of_physicians
Barbara Clark	1
Charles Robinson	1
David Rodriguez	1
Dorothy Thompson	1
Elizabeth Davis	1
Grace Wong	1
James Wilson	1
Jennifer Kim	1
John Smith	1
Joseph White	1
Linda Martinez	1
Margaret Harris	1
Maria Garcia	1
Michael Brown	1
Patricia Lee	1
Richard Moore	1
Robert Johnson	1

14

Write a query in SQL to count number of unique patients who got an appointment for examination room C.

```
SELECT COUNT(DISTINCT patient) AS  
unique_patients_in_room_C  
  
FROM appointment  
  
WHERE examinationroom = 105;
```

: unique\_patients\_in\_room\_C

4



15

Write a query in SQL to find the name of the patients and the number of the room where they have to go for their treatment.



```
SELECT p.name AS patient_name, COUNT(DISTINCT a.physician) AS  
number_of_physicians  
FROM patient p  
JOIN appointment a ON p.ssn = a.patient  
GROUP BY p.ssn, p.name  
ORDER BY number_of_physicians DESC, p.name;
```

name	examinationroom
John Smith	101
Grace Wong	102
Robert Johnson	103
Maria Garcia	104
James Wilson	105
Patricia Lee	101
Michael Brown	102
Linda Martinez	103
William Taylor	104
Elizabeth Davis	105
David Rodriguez	101
Jennifer Kim	102
Richard Moore	103
Susan Jackson	104
Joseph White	105
Margaret Harris	101
Thomas Martin	102



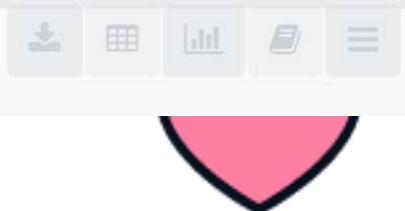
16

Write a query in SQL to find the name of the nurses and the room scheduled, where they will assist the physicians.



```
SELECT n.name AS nurse_name, a.examinationroom AS  
room_number  
FROM nurse n  
JOIN appointment a ON n.employeeid = a.prepurse;
```

nurse_name	room_number
Nurse Amy Pond	101
Nurse Rory Williams	102
Nurse Clara Oswald	103
Nurse Martha Jones	104
Nurse Donna Noble	105
Nurse Rose Tyler	101
Nurse River Song	102
Nurse Jackie Tyler	103
Nurse Sarah Jane Smith	104
Nurse Jo Grant	105
Nurse Liz Shaw	101
Nurse Leela	102
Nurse Romana	103
Nurse Nyssa	104
Nurse Tegan Jovanka	105
Nurse Peri Brown	101
Nurse Mel Bush	102



17

Write a query in SQL to find the name of the patients who taken the appointment on the 25th of April at 10 am, and also display their physician, assisting nurses and room no.

```
SELECT p.name AS patient_name, ph.name AS physician_name,  
n.name AS nurse_name, a.examinationroom AS room_number  
FROM appointment a  
JOIN patient p ON a.patient = p.ssn  
JOIN physician ph ON a.physician = ph.employeeid  
LEFT JOIN nurse n ON a.prepurse = n.employeeid  
WHERE date(a.start_dt_time) = '2025-04-25'  
AND strftime('%H', a.start_dt_time) = '10';
```

patient_name	physician_name	nurse_name	room_number
Grace Wong	Dr. Jane Doe	Nurse Rory Williams	102



18

Write a query in SQL to find the name of patients and their physicians who does not require any assistance of a nurse.

```
SELECT p.name AS patient_name, ph.name AS physician_name  
FROM appointment a  
JOIN patient p ON a.patient = p.ssn  
JOIN physician ph ON a.physician = ph.employeeid  
WHERE a.prepurse IS NULL;
```

patient_name	physician_name



19

Write a query in SQL to find the name of the patients, their treating physicians and medication



```
SELECT p.name AS patient_name, ph.name AS physician_name,  
m.name AS medication  
FROM prescribes pr  
JOIN patient p ON pr.patient = p.ssn  
JOIN physician ph ON pr.physician = ph.employeeid  
JOIN medication m ON pr.medication = m.code;
```

patient_name	physician_name	medication
John Smith	Dr. John Smith	Lisinopril
Grace Wong	Dr. Jane Doe	Metformin
Robert Johnson	Dr. Robert Johnson	Atorvastatin
Maria Garcia	Dr. Emily Williams	Levothyroxine
James Wilson	Dr. Michael Brown	Amlodipine
Patricia Lee	Dr. Sarah Davis	Metoprolol
Michael Brown	Dr. David Miller	Omeprazole
Linda Martinez	Dr. Jennifer Wilson	Losartan
William Taylor	Dr. Richard Moore	Albuterol
Elizabeth Davis	Dr. Lisa Taylor	Gabapentin
David Rodriguez	Dr. Thomas Anderson	Hydrochlorothiazide
Jennifer Kim	Dr. Nancy Thomas	Sertraline
Richard Moore	Dr. Daniel Jackson	Amoxicillin
Susan Jackson	Dr. Karen White	Prednisone
Joseph White	Dr. Christopher Harris	Fluticasone
Margaret Harris	Dr. Amanda Martin	Furosemide
Thomas Martin	Dr. Matthew Thompson	Insulin Glargine
patient_name	physician_name	medication
Dorothy Thompson	Dr. Stephanie Garcia	Warfarin
Charles Robinson	Dr. Kevin Martinez	Citalopram
Barbara Clark	Dr. Jessica Robinson	Azithromycin



20

Write a query in SQL to find the name of the patients who taken an advanced appointment, and also display their physicians and medication.



```
SELECT p.name AS patient_name, ph.name AS physician_name,  
m.name AS medication  
FROM appointment a  
JOIN patient p ON a.patient = p.ssn  
JOIN physician ph ON a.physician = ph.employeeid  
JOIN prescribes pr ON a.appointmentid = pr.appointment  
JOIN medication m ON pr.medication = m.code;
```

patient_name	physician_name	medication
John Smith	Dr. John Smith	Lisinopril
Grace Wong	Dr. Jane Doe	Metformin
Robert Johnson	Dr. Robert Johnson	Atorvastatin
Maria Garcia	Dr. Emily Williams	Levothyroxine
James Wilson	Dr. Michael Brown	Amlodipine
Patricia Lee	Dr. Sarah Davis	Metoprolol
Michael Brown	Dr. David Miller	Omeprazole
Linda Martinez	Dr. Jennifer Wilson	Losartan
William Taylor	Dr. Richard Moore	Albuterol
Elizabeth Davis	Dr. Lisa Taylor	Gabapentin
David Rodriguez	Dr. Thomas Anderson	Hydrochlorothiazide
Jennifer Kim	Dr. Nancy Thomas	Sertraline
Richard Moore	Dr. Daniel Jackson	Amoxicillin
Susan Jackson	Dr. Karen White	Prednisone
Joseph White	Dr. Christopher Harris	Fluticasone
Margaret Harris	Dr. Amanda Martin	Furosemide
Thomas Martin	Dr. Matthew Thompson	Insulin Glargine



21

Write a query in SQL to find the name and medication for those patients who did not take any appointment.

```
SELECT p.name AS patient_name, m.name AS medication  
FROM patient p  
JOIN prescribes pr ON p.ssn = pr.patient  
JOIN medication m ON pr.medication = m.code  
LEFT JOIN appointment a ON p.ssn = a.patient  
WHERE a.appointmentid IS NULL;
```

patient_name	medication



O \_ X

22

Write a query in SQL to count the number of available rooms in each block.

```
SELECT blockcode, COUNT(*) AS available_rooms  
FROM room  
WHERE unavailable = FALSE  
GROUP BY blockcode  
ORDER BY blockcode;
```

blockcode	available_rooms
101	2
102	1
201	2
202	1
301	1
302	2
401	1
402	2
501	2
502	1



23

Write a query in SQL to count the number of available rooms in each floor.

```
SELECT blockfloor, COUNT(*) AS available_rooms  
FROM room  
WHERE unavailable = FALSE  
GROUP BY blockfloor  
ORDER BY blockfloor;
```

blockfloor	available_rooms
1	3
2	3
3	3
4	3
5	3



O \_ X

24

Write a query in SQL to count the number of available rooms for each block in each floor.

```
SELECT blockfloor, COUNT(*) AS available_rooms  
FROM room  
WHERE unavailable = FALSE  
GROUP BY blockfloor  
ORDER BY blockfloor;
```

blockfloor	blockcode	available_rooms
1	101	2
1	102	1
2	201	2
2	202	1
3	301	1
3	302	2
4	401	1
4	402	2
5	501	2
5	502	1



O \_ X

25

Write a query in SQL to count the number of unavailable rooms for each block in each floor.

```
SELECT blockfloor, blockcode, COUNT(*) AS unavailable_rooms  
FROM room  
WHERE unavailable = TRUE  
GROUP BY blockfloor, blockcode  
ORDER BY blockfloor, blockcode;
```

blockfloor	blockcode	unavailable_rooms
1	102	1
2	202	1
3	301	1
4	401	1
5	502	1



26

Write a query in SQL to find out the floor where the maximum no of rooms are available.

```
SELECT blockfloor, COUNT(*) AS available_rooms  
FROM room  
WHERE unavailable = FALSE  
GROUP BY blockfloor  
ORDER BY available_rooms DESC  
LIMIT 1;
```

blockfloor	available_rooms
5	3



27

Write a query in SQL to find out the floor where the minimum no of rooms are available

```
SELECT blockfloor, COUNT(*) AS available_rooms  
FROM room  
WHERE unavailable = FALSE  
GROUP BY blockfloor  
ORDER BY available_rooms ASC  
LIMIT 1;
```

blockfloor	available_rooms
1	3



28

Write a query in SQL to obtain the name of the patients, their block, floor, and room number where they are admitted.



```
SELECT p.name AS patient_name, r.blockfloor, r.blockcode,  
r.roomnumber  
FROM stay s  
JOIN patient p ON s.patient = p.ssn  
JOIN room r ON s.room = r.roomnumber;
```

patient_name	blockfloor	blockcode	roomnumber
John Smith	1	101	101
Grace Wong	1	101	102
Robert Johnson	1	102	103
Maria Garcia	2	201	201
James Wilson	2	201	202
Patricia Lee	2	202	203
Michael Brown	3	301	301
Linda Martinez	3	301	302
William Taylor	3	302	303
Elizabeth Davis	1	101	101
David Rodriguez	1	101	102
Jennifer Kim	1	102	103
Richard Moore	2	201	201
Susan Jackson	2	201	202
Joseph White	2	202	203
Margaret Harris	3	301	301
Thomas Martin	3	301	302



29

Write a query in SQL to obtain the nurses and the block where they are booked for attending the patients on call.



```
SELECT n.name AS nurse_name, o.blockfloor, o.blockcode  
FROM on_call o  
JOIN nurse n ON o.nurse = n.employeeid;
```

nurse_name	blockfloor	blockcode
Nurse Amy Pond	1	101
Nurse Rory Williams	2	102
Nurse Clara Oswald	3	103
Nurse Martha Jones	1	201
Nurse Donna Noble	2	202
Nurse Rose Tyler	3	203
Nurse River Song	1	301
Nurse Jackie Tyler	2	302
Nurse Sarah Jane Smith	3	303
Nurse Jo Grant	1	101
Nurse Amy Pond	2	102
Nurse Rory Williams	3	103
Nurse Clara Oswald	1	201
Nurse Martha Jones	2	202
Nurse Donna Noble	3	203
Nurse Rose Tyler	1	301
Nurse River Song	2	302



30

Write a query in SQL to make a report which will show -  
a) name of the patient,  
b) name of the physician who is treating him or her,  
c) name of the nurse who is attending him or her,  
d) which treatment is going on to the patient,  
e) the date of release,  
f) in which room the patient has admitted and which floor and block the room belongs to respectively.



SELECT

```
p.name AS patient_name,  
ph.name AS treating_physician,  
n.name AS attending_nurse,  
pr.name AS treatment,  
s.end_time AS release_date,  
r.roomnumber AS room_number,  
r.blockfloor AS floor,  
r.blockcode AS block
```

FROM stay s

```
JOIN patient p ON s.patient = p.ssn  
JOIN room r ON s.room = r.roomnumber  
LEFT JOIN undergoes u ON s.stayid = u.stay  
LEFT JOIN procedure pr ON u.procedure = pr.code  
LEFT JOIN physician ph ON u.physician = ph.employeeid  
LEFT JOIN nurse n ON u.assistingnurse = n.employeeid;
```



30

Write a query in SQL to make a report which will show -  
a) name of the patient,  
b) name of the physician who is treating him or her,  
c) name of the nurse who is attending him or her,  
d) which treatment is going on to the patient,  
e) the date of release,  
f) in which room the patient has admitted and which floor and block the room belongs to respectively.



#	patient_id	treating_physician	attending_nurse	treatment	release_date	room_number	floor	block
1	John Smith	Dr. John Smith	Nurse Amy Johnson	Appendectomy	2025-01-01	101	1	101
2	Grace Wong	Dr. Jane Doe	Nurse Rosemary	Coronary Bypass	2025-01-02	102	1	101
3	Robert Johnson	Dr. Robert Lee	Nurse Clara	CT Scan	2025-01-03	103	1	102
4	Maria Garcia	Dr. Emily White	Nurse Margaret	MRI	2025-01-04	201	2	201
5	James Wilson	Dr. Michael Green	Nurse Donald	X-Ray	2025-01-05	202	2	201
6	Patricia Lee	Dr. Sarah Black	Nurse Rose	Appendectomy	2025-01-06	203	2	202
7	Michael Brown	Dr. David Blue	Nurse River	Coronary Bypass	2025-01-07	301	3	301
8	Linda Martinez	Dr. Jennifer White	Nurse Jacob	CT Scan	2025-01-08	302	3	301
9	William Taylor	Dr. Richard Grey	Nurse Sarah	MRI	2025-01-09	303	3	302
10	Elizabeth Davis	Dr. Lisa Taylor	Nurse Jordan	X-Ray	2025-01-10	101	1	101
11	David Rodriguez	Dr. John Smith	Nurse Amy Johnson	Appendectomy	2025-01-11	102	1	101
12	Jennifer Kim	Dr. Jane Doe	Nurse Rosemary	Coronary Bypass	2025-01-12	103	1	102
13	Richard Martin	Dr. Robert Lee	Nurse Clara	CT Scan	2025-01-13	201	2	201
14	Susan Jackson	Dr. Emily White	Nurse Margaret	MRI	2025-01-14	202	2	201
15	Joseph Wilson	Dr. Michael Green	Nurse Donald	X-Ray	2025-01-15	203	2	202
16	Margaret Lee	Dr. Sarah Black	Nurse Rose	Appendectomy	2025-01-16	301	3	301
17	Thomas Martin	Dr. David Blue	Nurse River	Coronary Bypass	2025-01-17	302	3	301
18	Dorothy Thomas	Dr. Jennifer White	Nurse Jacob	CT Scan	2025-01-18	303	3	302
19	Charles Rodriguez	Dr. Richard Grey	Nurse Sarah	MRI	2025-01-19	101	1	101
20	Barbara Clark	Dr. Lisa Taylor	Nurse Jordan	X-Ray	2025-01-20	102	1	101

31

Write a SQL query to obtain the names of all the physicians performed a medical procedure but they are not ceritified to perform.



```
SELECT DISTINCT ph.name AS physician_name  
FROM undergoes u  
JOIN physician ph ON u.physician = ph.employeeid  
LEFT JOIN trained_in ti ON u.physician = ti.physician AND  
u.procedure = ti.treatment  
WHERE ti.physician IS NULL;
```

physician\_name

Dr. Sarah Davis

Dr. David Miller

Dr. Jennifer Wilson

Dr. Richard Moore

Dr. Lisa Taylor



32

Write a query in SQL to obtain the names of all the physicians, their procedure, date when the procedure was carried out and name of the patient on which procedure have been carried out but those physicians are not certified for that procedure.



SELECT

```
ph.name AS physician_name,  
pr.name AS procedure_name,  
u.date AS procedure_date,  
p.name AS patient_name  
FROM undergoes u  
JOIN physician ph ON u.physician = ph.employeeid  
JOIN procedure pr ON u.procedure = pr.code  
JOIN patient p ON u.patient = p.ssn  
LEFT JOIN trained_in ti ON u.physician = ti.physician AND  
u.procedure = ti.treatment  
WHERE ti.physician IS NULL;
```

i	physician_name	procedure_name	procedure_date	patient_name
	Dr. Sarah Davis	Appendectomy	2025-01-06	Patricia Lee
	Dr. David Miller	Coronary Bypass	2025-01-07	Michael Brown
	Dr. Jennifer Wilson	CT Scan	2025-01-08	Linda Martinez
	Dr. Richard Moore	MRI	2025-01-09	William Taylor
	Dr. Lisa Taylor	X-Ray	2025-01-10	Elizabeth Davis
	Dr. Sarah Davis	Appendectomy	2025-01-16	Margaret Harris
	Dr. David Miller	Coronary Bypass	2025-01-17	Thomas Martin
	Dr. Jennifer Wilson	CT Scan	2025-01-18	Dorothy Thompson
	Dr. Richard Moore	MRI	2025-01-19	Charles Robinson
	Dr. Lisa Taylor	X-Ray	2025-01-20	Barbara Clark



33

Write a query in SQL to obtain the name and position of all physicians who completed a medical procedure with certification after the date of expiration of their certificate.

```
SELECT DISTINCT p.name AS physician_name, p.position  
FROM undergoes u  
JOIN physician p ON u.physician = p.employeeid  
JOIN trained_in ti ON u.physician = ti.physician AND u.procedure =  
ti.treatment  
WHERE u.date > ti.certificationexpires;
```

physician_name	position



34

Write a query in SQL to obtain the name of all those physicians who completed a medical procedure with certification after the date of expiration of their certificate, their position, procedure they have done, date of procedure, name of the patient on which the procedure had been applied and the date when the certification expired.

SELECT

```
p.name AS physician_name,  
p.position,  
pr.name AS procedure_name,  
u.date AS procedure_date,  
pt.name AS patient_name,  
ti.certificationexpires AS certification_expiry_date,  
julianday(ti.certificationexpires) - julianday(u.date) AS  
days_until_expiry
```

FROM undergoes u

JOIN physician p ON u.physician = p.employeeid

JOIN procedure pr ON u.procedure = pr.code

JOIN patient pt ON u.patient = pt.ssn

JOIN trained\_in ti ON u.physician = ti.physician AND u.procedure =  
ti.treatment

WHERE u.date &gt; ti.certificationexpires;

physicia...	position	procedure_...	procedure_...	patient_name	certificatio...	days_until_expiry
-------------	----------	---------------	---------------	--------------	-----------------	-------------------



35

Write a query in SQL to obtain the names of all the nurses who have ever been on call for room 101.

```
SELECT DISTINCT n.name AS nurse_name  
FROM on_call o  
JOIN nurse n ON o.nurse = n.employeeid  
JOIN room r ON o.blockfloor = r.blockfloor AND o.blockcode =  
r.blockcode  
WHERE r.roomnumber = 101;
```

⋮ nurse\_name

Nurse Amy Pond

Nurse Jo Grant

Nurse Sarah Jane Smith



36

Write a query in SQL to Obtain the names of all patients who has been prescribed some medication by his/her physician who has carried out primarycare and the name of that physician.



```
SELECT p.name AS patient_name, ph.name AS  
primary_care_physician  
FROM prescribes pr  
JOIN patient p ON pr.patient = p.ssn  
JOIN physician ph ON pr.physician = ph.employeeid  
WHERE pr.physician = p.pcp;
```

patient_name	primary_care_physician
John Smith	Dr. John Smith
Grace Wong	Dr. Jane Doe
Robert Johnson	Dr. Robert Johnson
Maria Garcia	Dr. Emily Williams
James Wilson	Dr. Michael Brown
Patricia Lee	Dr. Sarah Davis
Michael Brown	Dr. David Miller
Linda Martinez	Dr. Jennifer Wilson
William Taylor	Dr. Richard Moore
Elizabeth Davis	Dr. Lisa Taylor
David Rodriguez	Dr. Thomas Anderson
Jennifer Kim	Dr. Nancy Thomas
Richard Moore	Dr. Daniel Jackson
Susan Jackson	Dr. Karen White
Joseph White	Dr. Christopher Harris
Margaret Harris	Dr. Amanda Martin
Thomas Martin	Dr. Matthew Thompson



37

Write a query in SQL to obtain the names of all patients who has been undergone a procedure costing more than \$5,000 and the name of that physician who has carried out primary care.



```
SELECT p.name AS patient_name, ph.name AS  
primary_care_physician  
FROM undergoes u  
JOIN procedure pr ON u.procedure = pr.code  
JOIN patient p ON u.patient = p.ssn  
JOIN physician ph ON p.pcp = ph.employeeid  
WHERE pr.cost > 5000;
```

patient_name	primary_care_physician
John Smith	Dr. John Smith
Grace Wong	Dr. Jane Doe
Patricia Lee	Dr. Sarah Davis
Michael Brown	Dr. David Miller
David Rodriguez	Dr. Thomas Anderson
Jennifer Kim	Dr. Nancy Thomas
Margaret Harris	Dr. Amanda Martin
Thomas Martin	Dr. Matthew Thompson



38

Write a query in SQL to Obtain the names of all patients who had at least two appointment where the nurse who prepped the appointment was a registered nurse and the physician who has carried out primary care.



```
SELECT p.name AS patient_name  
FROM appointment a  
JOIN patient p ON a.patient = p.ssn  
JOIN nurse n ON a.prepurse = n.employeeid  
WHERE n.registered = TRUE;
```

patient\_name

John Smith

Robert Johnson

James Wilson

Patricia Lee

Michael Brown

William Taylor

David Rodriguez

Richard Moore

Susan Jackson

Joseph White

Thomas Martin

Charles Robinson



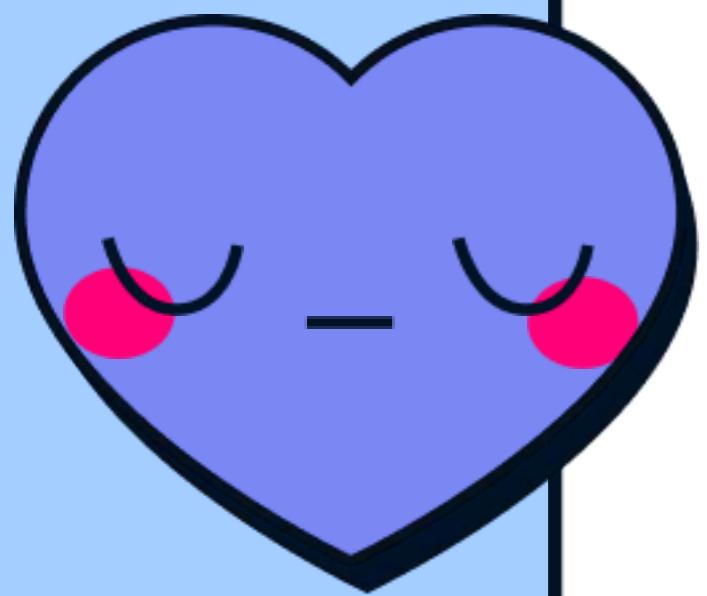
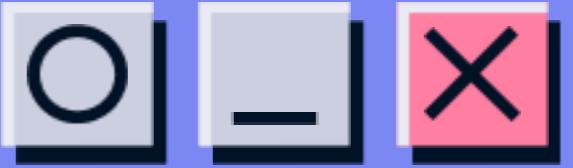
39

Write a query in SQL to Obtain the names of all patients whose primary care is taken by a physician who is not the head of any department and name of that physician along with their primary care physician.

```
SELECT p.name AS patient_name, ph.name AS  
primary_care_physician  
FROM patient p  
JOIN physician ph ON p.pcp = ph.employeeid  
LEFT JOIN department d ON ph.employeeid = d.head  
WHERE d.head IS NULL;
```

patient_name	primary_care_physician





# Thank you

