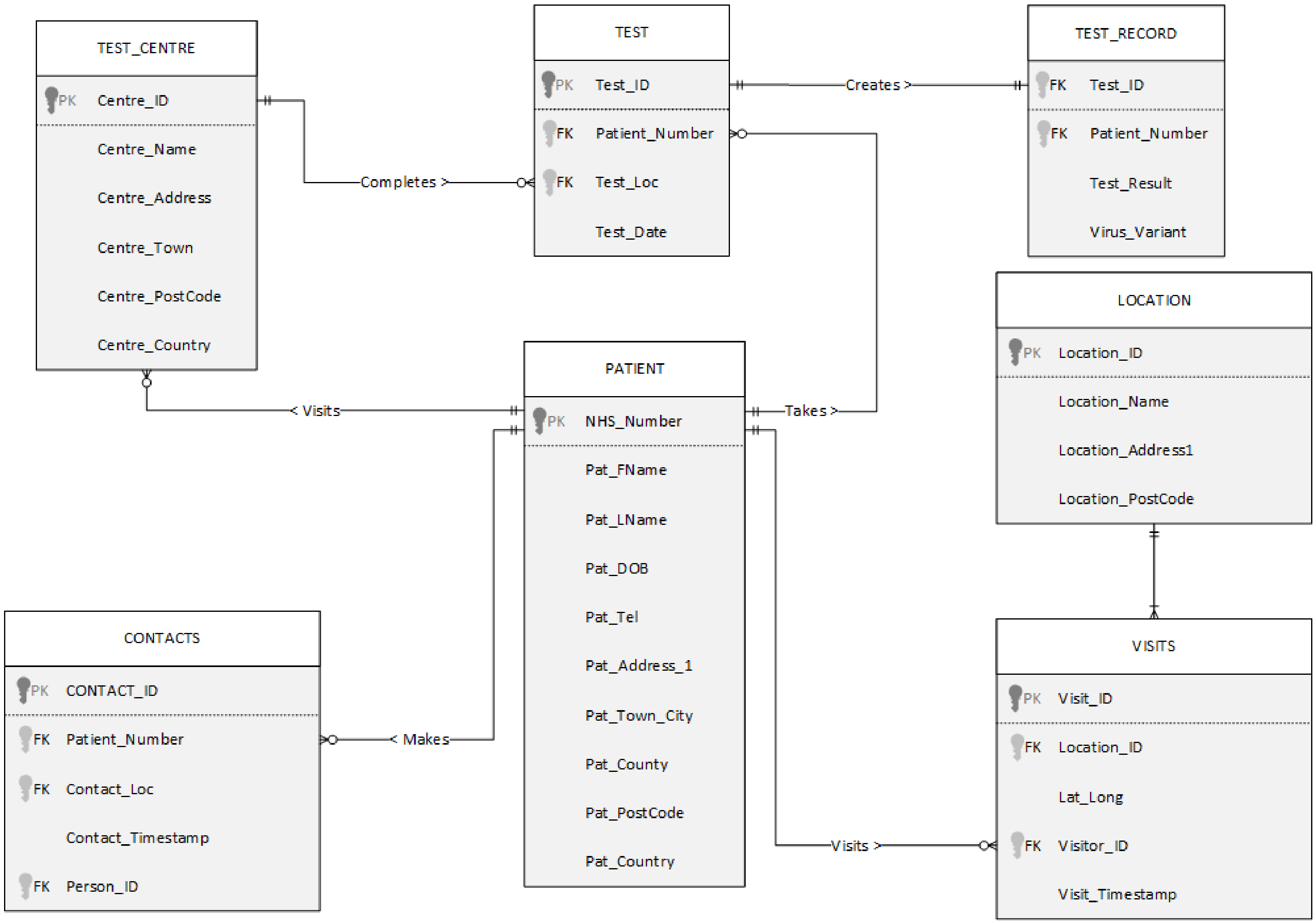
AUGUST 17, 2021

DATABASE DESIGN AND IMPLEMENTATION

BRIAN COLLINS
2078876

Task 1: Entity Relationship Diagram



ERD Explanation:

1. Patient entries are taken from the NHS database. The primary key is the patients unique 10-digit NHS Number.

2. A patient may attend test centres throughout the pandemic for testing. Not all patients will attend a test centre, and some patients will attend more than one test centre.

3. A patient may take a lateral flow test directly without attending a test centre. As such, the test location field for some test entries may contain null values. Furthermore, not all patients will take a test. Test records will hold details of patient NHS number (FK) and testing location (FK)(if applicable).

4. Test Centres may complete tests on zero or more patients, it is plausible that a test centre might complete no tests. A completed test record will be associated with a maximum of one test centre.

5. On analysis, a test will create an associated test record containing details of the test, and patient details will be referenced as the foreign key. All tests will have an associated test record.

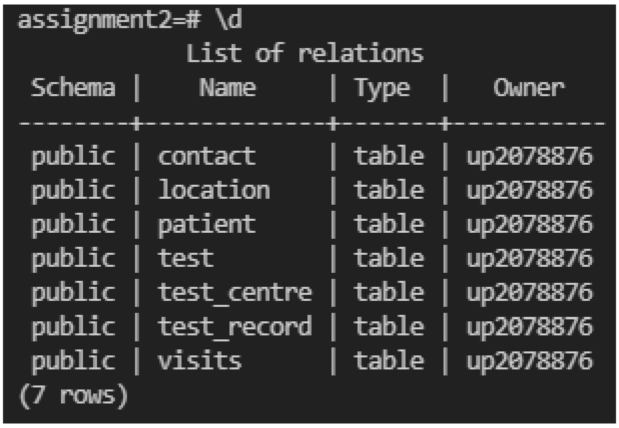
6. A patient can check-in to locations over the course of the pandemic, not all locations are formally recognised and with an associated have a static address, therefore latitude and longitude are used as a point value in the database to illustrate geographic placement. Patients may not check in to any locations over the course of the pandemic.

7. Any locations with a static address (formally recognised businesses) are automatically fed from the separate Locations table, using Location_ID as a foreign key in visits table. A location may be referenced by multiple separate visits, but a visit can only be to one location.

8. A patient may make contacts over the course of the pandemic. Not all contacts will have a location outlined in the visit table, therefore, this field can hold null values. Not all patients will come into contact with other people. Patient_Number is the PK of the patient making contact, Person_ID acts as a foreign key pointing back towards NHS_Number in the Patients table, acting as the contacted party.

Task 2: Database Creation in PostgreSQL

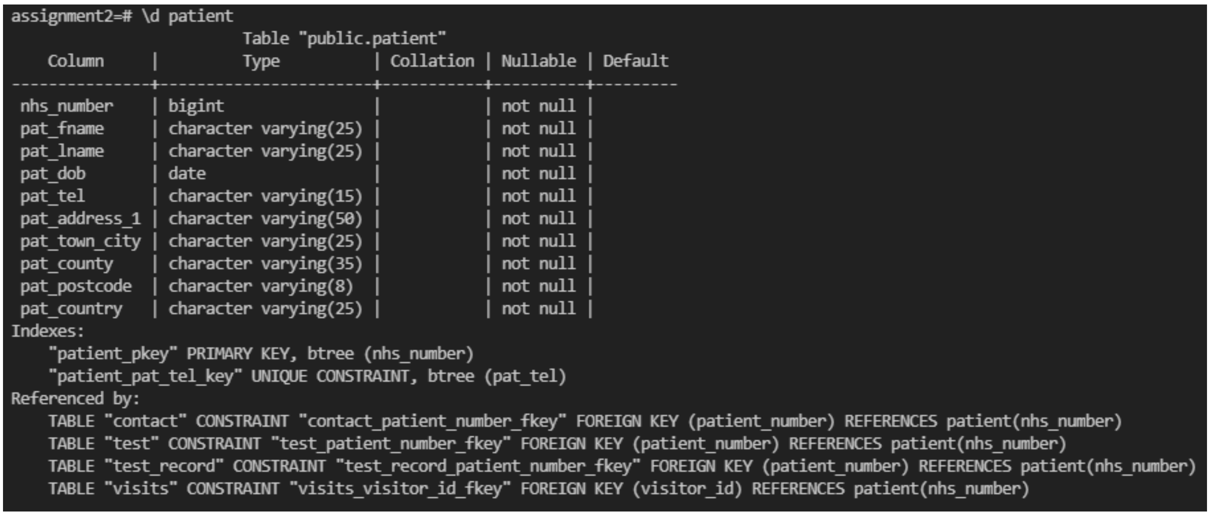
All tables:



```
assignment2=# \d
           List of relations
 Schema |    Name     | Type  |   Owner
--------+-------------+-------+-----------
 public | contact     | table | up2078876
 public | location    | table | up2078876
 public | patient     | table | up2078876
 public | test        | table | up2078876
 public | test_centre | table | up2078876
 public | test_record | table | up2078876
 public | visits      | table | up2078876
(7 rows)
```

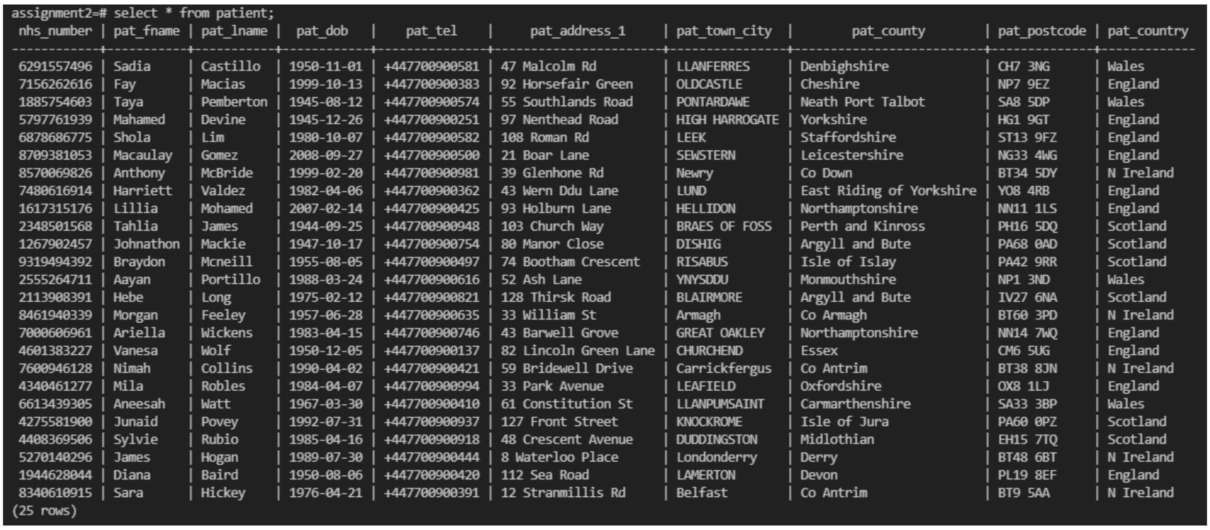
Patient:

Description:



```
assignment2=# \d patient
                        Table "public.patient"
    Column     |         Type          | Collation | Nullable | Default
---------------+-----------------------+-----------+----------+---------
 nhs_number    | bigint                |           | not null |
 pat_fname     | character varying(25) |           | not null |
 pat_lname     | character varying(25) |           | not null |
 pat_dob       | date                  |           | not null |
 pat_tel       | character varying(15) |           | not null |
 pat_address_1 | character varying(50) |           | not null |
 pat_town_city | character varying(25) |           | not null |
 pat_county    | character varying(35) |           | not null |
 pat_postcode  | character varying(8)  |           | not null |
 pat_country   | character varying(25) |           | not null |
Indexes:
    "patient_pkey" PRIMARY KEY, btree (nhs_number)
    "patient_pat_tel_key" UNIQUE CONSTRAINT, btree (pat_tel)
Referenced by:
    TABLE "contact" CONSTRAINT "contact_patient_number_fkey" FOREIGN KEY (patient_number) REFERENCES patient(nhs_number)
    TABLE "test" CONSTRAINT "test_patient_number_fkey" FOREIGN KEY (patient_number) REFERENCES patient(nhs_number)
    TABLE "test_record" CONSTRAINT "test_record_patient_number_fkey" FOREIGN KEY (patient_number) REFERENCES patient(nhs_number)
    TABLE "visits" CONSTRAINT "visits_visitor_id_fkey" FOREIGN KEY (visitor_id) REFERENCES patient(nhs_number)
```

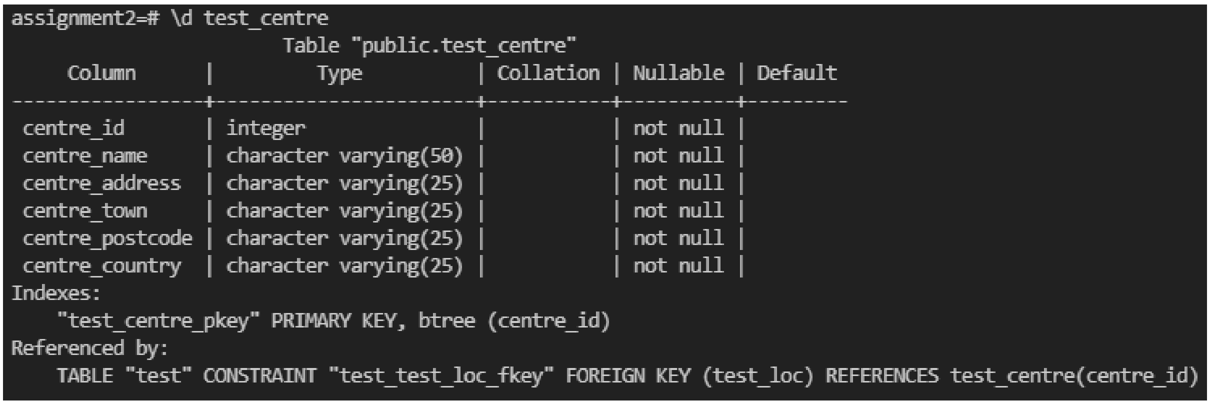
Dummy Data:



```
assignment2=# select * from patient;
 nhs_number | pat_fname | pat_lname |  pat_dob   |    pat_tel    |    pat_address_1     | pat_town_city  |        pat_county        | pat_postcode | pat_country
------------+-----------+-----------+------------+---------------+----------------------+----------------+--------------------------+--------------+-------------
 6291557496 | Sadia     | Castillo  | 1950-11-01 | +447700900581 | 47 Malcolm Rd        | LLANFERRES     | Denbighshire             | CH7 3NG      | Wales
 7156262616 | Fay       | Macias    | 1999-10-13 | +447700900383 | 92 Horsefair Green   | OLDCASTLE      | Cheshire                 | NP7 9EZ      | England
 1885754603 | Taya      | Pemberton | 1945-08-12 | +447700900574 | 55 Southlands Road   | PONTARDAWE     | Neath Port Talbot        | SA8 5DP      | Wales
 5797761939 | Mahamed   | Devine    | 1945-12-26 | +447700900251 | 97 Nenthead Road     | HIGH HARROGATE | Yorkshire                | HG1 9GT      | England
 6878686775 | Shola     | Lim       | 1980-10-07 | +447700900582 | 108 Roman Rd         | LEEK           | Staffordshire            | ST13 9FZ     | England
 8709381053 | Macaulay  | Gomez     | 2008-09-27 | +447700900500 | 21 Boar Lane         | SEWSTERN       | Leicestershire           | NG33 4WG     | England
 8570069826 | Anthony   | McBride   | 1999-02-20 | +447700900981 | 39 Glenhone Rd       | Newry          | Co Down                  | BT34 5DY     | N Ireland
 7480616914 | Harriett  | Valdez    | 1982-04-06 | +447700900362 | 43 Wern Ddu Lane     | LUND           | East Riding of Yorkshire | YO8 4RB      | England
 1617315176 | Lillia    | Mohamed   | 2007-02-14 | +447700900425 | 93 Holburn Lane      | HELLIDON       | Northamptonshire         | NN11 1LS     | England
 2348501568 | Tahlia    | James     | 1944-09-25 | +447700900948 | 103 Church Way       | BRAES OF FOSS  | Perth and Kinross        | PH16 5DQ     | Scotland
 1267902457 | Johnathon | Mackie    | 1947-10-17 | +447700900754 | 80 Manor Close       | DISHIG         | Argyll and Bute          | PA68 0AD     | Scotland
 9319494392 | Braydon   | Mcneill   | 1955-08-05 | +447700900497 | 74 Bootham Crescent  | RISABUS        | Isle of Islay            | PA42 9RR     | Scotland
 2555264711 | Aayan     | Portillo  | 1988-03-24 | +447700900616 | 52 Ash Lane          | YNYSDDU        | Monmouthshire            | NP1 3ND      | Wales
 2113908391 | Hebe      | Long      | 1975-02-12 | +447700900821 | 128 Thirsk Road      | BLAIRMORE      | Argyll and Bute          | IV27 6NA     | Scotland
 8461940339 | Morgan    | Feeley    | 1957-06-28 | +447700900635 | 33 William St        | Armagh         | Co Armagh                | BT60 3PD     | N Ireland
 7000606961 | Ariella   | Wickens   | 1983-04-15 | +447700900746 | 43 Barwell Grove     | GREAT OAKLEY   | Northamptonshire         | NN14 7WQ     | England
 4601383227 | Vanesa    | Wolf      | 1950-12-05 | +447700900137 | 82 Lincoln Green Lane | CHURCHEND      | Essex                    | CM6 5UG      | England
 7600946128 | Nimah     | Collins   | 1990-04-02 | +447700900421 | 59 Bridewell Drive   | Carrickfergus  | Co Antrim                | BT38 8JN     | N Ireland
 4340461277 | Mila      | Robles    | 1984-04-07 | +447700900994 | 33 Park Avenue       | LEAFIELD       | Oxfordshire              | OX8 1LJ      | England
 6613439305 | Aneesah   | Watt      | 1967-03-30 | +447700900410 | 61 Constitution St   | LLANPUMSAINT   | Carmarthenshire          | SA33 3BP     | Wales
 4275581900 | Junaid    | Povey     | 1992-07-31 | +447700900937 | 127 Front Street     | KNOCKROME      | Isle of Jura             | PA60 0PZ     | Scotland
 4408369506 | Sylvie    | Rubio     | 1985-04-16 | +447700900918 | 48 Crescent Avenue   | DUDDINGSTON    | Midlothian               | EH15 7TQ     | Scotland
 5270140296 | James     | Hogan     | 1989-07-30 | +447700900444 | 8 Waterloo Place     | Londonderry    | Derry                    | BT48 6BT     | N Ireland
 1944628044 | Diana     | Baird     | 1950-08-06 | +447700900420 | 112 Sea Road         | LAMERTON       | Devon                    | PL19 8EF     | England
 8340610915 | Sara      | Hickey    | 1976-04-21 | +447700900391 | 12 Stranmillis Rd    | Belfast        | Co Antrim                | BT9 5AA      | N Ireland
(25 rows)
```

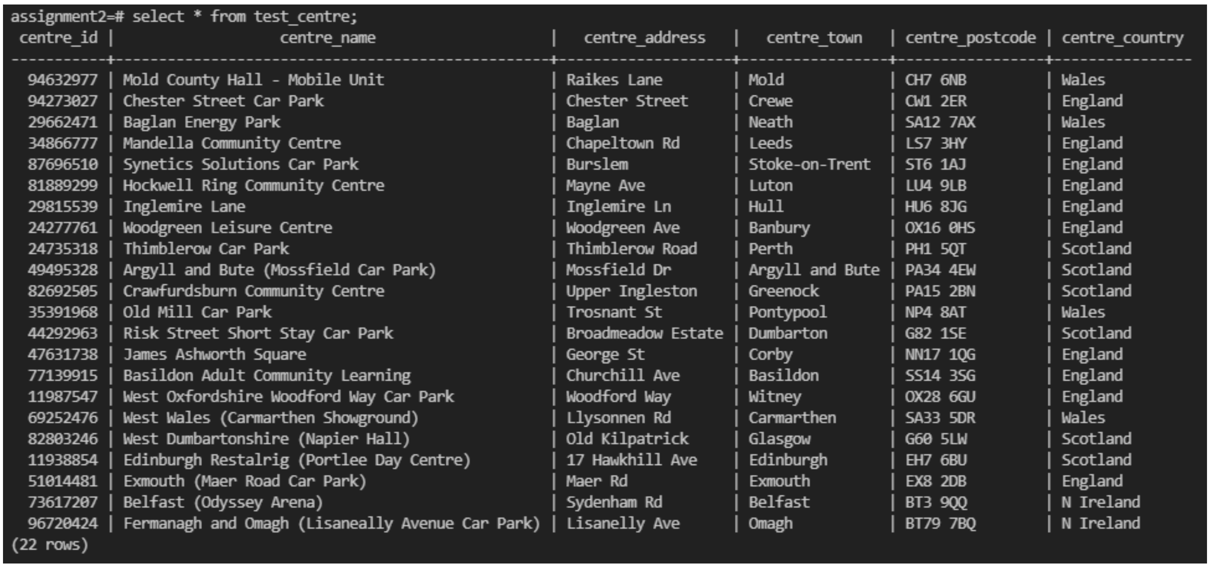
Test Centre:

Description:



```
assignment2=# \d test_centre
                      Table "public.test_centre"
     Column      |         Type          | Collation | Nullable | Default
-----------------+-----------------------+-----------+----------+---------
 centre_id       | integer               |           | not null |
 centre_name     | character varying(50) |           | not null |
 centre_address  | character varying(25) |           | not null |
 centre_town     | character varying(25) |           | not null |
 centre_postcode | character varying(25) |           | not null |
 centre_country  | character varying(25) |           | not null |
Indexes:
    "test_centre_pkey" PRIMARY KEY, btree (centre_id)
Referenced by:
    TABLE "test" CONSTRAINT "test_test_loc_fkey" FOREIGN KEY (test_loc) REFERENCES test_centre(centre_id)
```

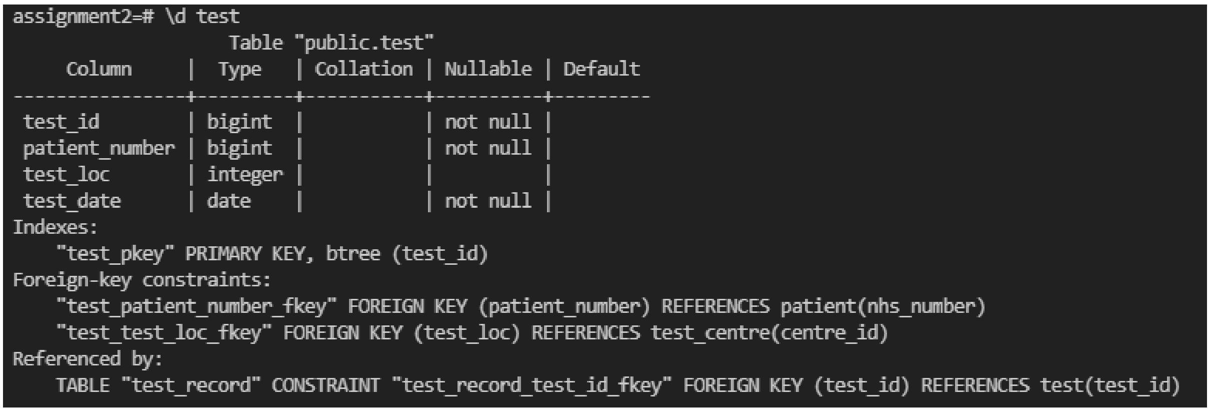
Dummy Data:



```
assignment2=# select * from test_centre;
 centre_id |                   centre_name                    |   centre_address   |   centre_town   | centre_postcode | centre_country
-----------+--------------------------------------------------+--------------------+-----------------+-----------------+----------------
  94632977 | Mold County Hall - Mobile Unit                   | Raikes Lane        | Mold            | CH7 6NB         | Wales
  94273027 | Chester Street Car Park                          | Chester Street     | Crewe           | CW1 2ER         | England
  29662471 | Baglan Energy Park                               | Baglan             | Neath           | SA12 7AX        | Wales
  34866777 | Mandella Community Centre                        | Chapeltown Rd      | Leeds           | LS7 3HY         | England
  87696510 | Synetics Solutions Car Park                      | Burslem            | Stoke-on-Trent  | ST6 1AJ         | England
  81889299 | Hockwell Ring Community Centre                   | Mayne Ave          | Luton           | LU4 9LB         | England
  29815539 | Inglemire Lane                                   | Inglemire Ln       | Hull            | HU6 8JG         | England
  24277761 | Woodgreen Leisure Centre                         | Woodgreen Ave      | Banbury         | OX16 0HS        | England
  24735318 | Thimblerow Car Park                              | Thimblerow Road    | Perth           | PH1 5QT         | Scotland
  49495328 | Argyll and Bute (Mossfield Car Park)             | Mossfield Dr       | Argyll and Bute | PA34 4EW        | Scotland
  82692505 | Crawfurdsburn Community Centre                   | Upper Ingleston    | Greenock        | PA15 2BN        | Scotland
  35391968 | Old Mill Car Park                                | Trosnant St        | Pontypool       | NP4 8AT         | Wales
  44292963 | Risk Street Short Stay Car Park                  | Broadmeadow Estate | Dumbarton       | G82 1SE         | Scotland
  47631738 | James Ashworth Square                            | George St          | Corby           | NN17 1QG        | England
  77139915 | Basildon Adult Community Learning                | Churchill Ave      | Basildon        | SS14 3SG        | England
  11987547 | West Oxfordshire Woodford Way Car Park           | Woodford Way       | Witney          | OX28 6GU        | England
  69252476 | West Wales (Carmarthen Showground)               | Llysonnen Rd       | Carmarthen      | SA33 5DR        | Wales
  82803246 | West Dumbartonshire (Napier Hall)                | Old Kilpatrick     | Glasgow         | G60 5LW         | Scotland
  11938854 | Edinburgh Restalrig (Portlee Day Centre)         | 17 Hawkhill Ave    | Edinburgh       | EH7 6BU         | Scotland
  51014481 | Exmouth (Maer Road Car Park)                     | Maer Rd            | Exmouth         | EX8 2DB         | England
  73617207 | Belfast (Odyssey Arena)                          | Sydenham Rd        | Belfast         | BT3 9QQ         | N Ireland
  96720424 | Fermanagh and Omagh (Lisaneally Avenue Car Park) | Lisanelly Ave      | Omagh           | BT79 7BQ        | N Ireland
(22 rows)
```

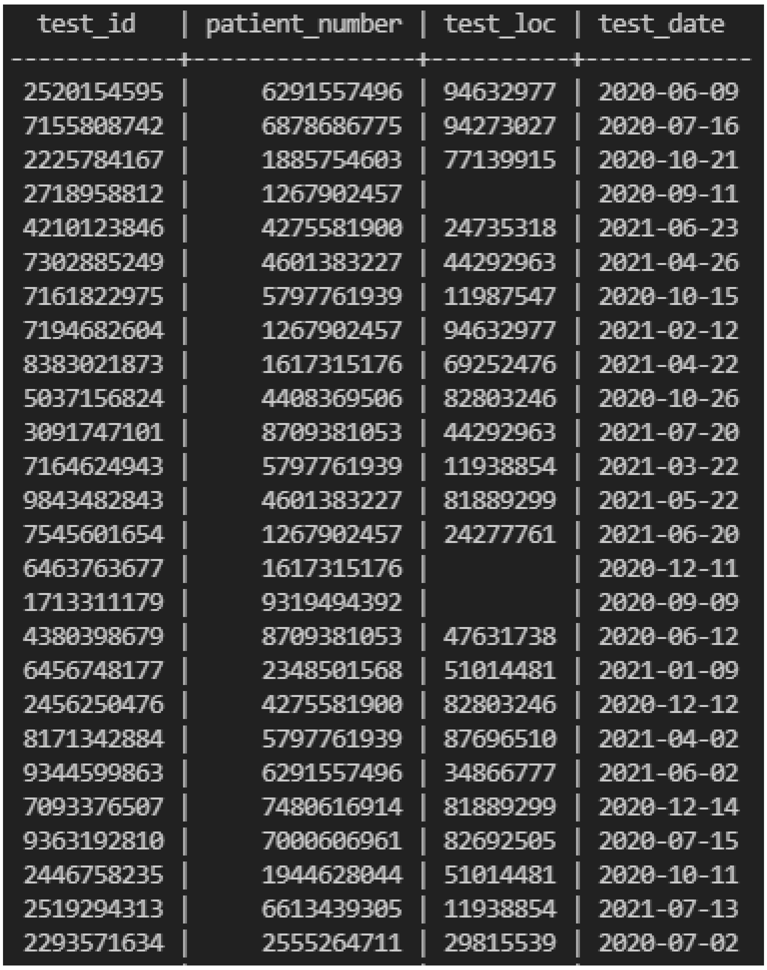
Test:

Description:



```
assignment2=# \d test
                  Table "public.test"
     Column     |  Type   | Collation | Nullable | Default
----------------+---------+-----------+----------+---------
 test_id        | bigint  |           | not null |
 patient_number | bigint  |           | not null |
 test_loc       | integer |           |          |
 test_date      | date    |           | not null |
Indexes:
    "test_pkey" PRIMARY KEY, btree (test_id)
Foreign-key constraints:
    "test_patient_number_fkey" FOREIGN KEY (patient_number) REFERENCES patient(nhs_number)
    "test_test_loc_fkey" FOREIGN KEY (test_loc) REFERENCES test_centre(centre_id)
Referenced by:
    TABLE "test_record" CONSTRAINT "test_record_test_id_fkey" FOREIGN KEY (test_id) REFERENCES test(test_id)
```

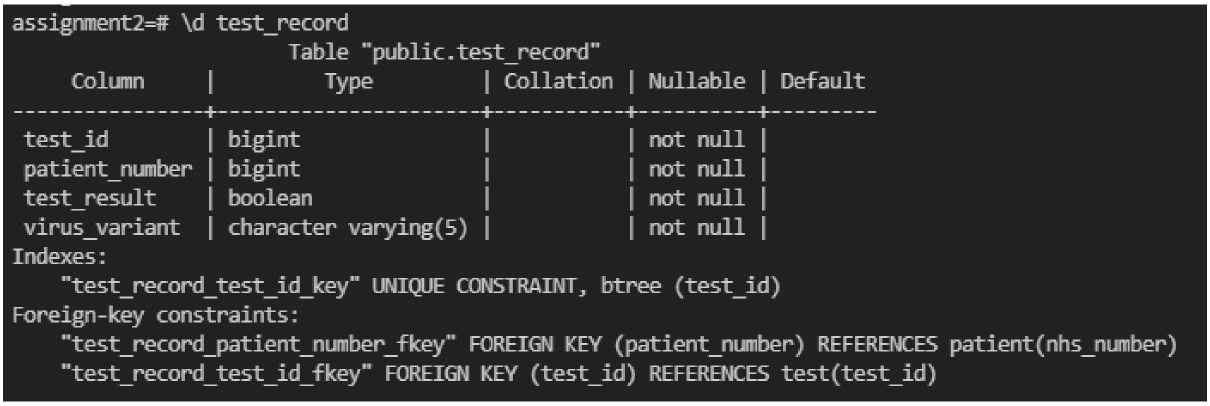
Dummy Data (Partial):



```
  test_id   | patient_number | test_loc | test_date
------------+----------------+----------+------------
 2520154595 |     6291557496 | 94632977 | 2020-06-09
 7155808742 |     6878686775 | 94273027 | 2020-07-16
 2225784167 |     1885754603 | 77139915 | 2020-10-21
 2718958812 |     1267902457 |          | 2020-09-11
 4210123846 |     4275581900 | 24735318 | 2021-06-23
 7302885249 |     4601383227 | 44292963 | 2021-04-26
 7161822975 |     5797761939 | 11987547 | 2020-10-15
 7194682604 |     1267902457 | 94632977 | 2021-02-12
 8383021873 |     1617315176 | 69252476 | 2021-04-22
 5037156824 |     4408369506 | 82803246 | 2020-10-26
 3091747101 |     8709381053 | 44292963 | 2021-07-20
 7164624943 |     5797761939 | 11938854 | 2021-03-22
 9843482843 |     4601383227 | 81889299 | 2021-05-22
 7545601654 |     1267902457 | 24277761 | 2021-06-20
 6463763677 |     1617315176 |          | 2020-12-11
 1713311179 |     9319494392 |          | 2020-09-09
 4380398679 |     8709381053 | 47631738 | 2020-06-12
 6456748177 |     2348501568 | 51014481 | 2021-01-09
 2456250476 |     4275581900 | 82803246 | 2020-12-12
 8171342884 |     5797761939 | 87696510 | 2021-04-02
 9344599863 |     6291557496 | 34866777 | 2021-06-02
 7093376507 |     7480616914 | 81889299 | 2020-12-14
 9363192810 |     7000606961 | 82692505 | 2020-07-15
 2446758235 |     1944628044 | 51014481 | 2020-10-11
 2519294313 |     6613439305 | 11938854 | 2021-07-13
 2293571634 |     2555264711 | 29815539 | 2020-07-02
```

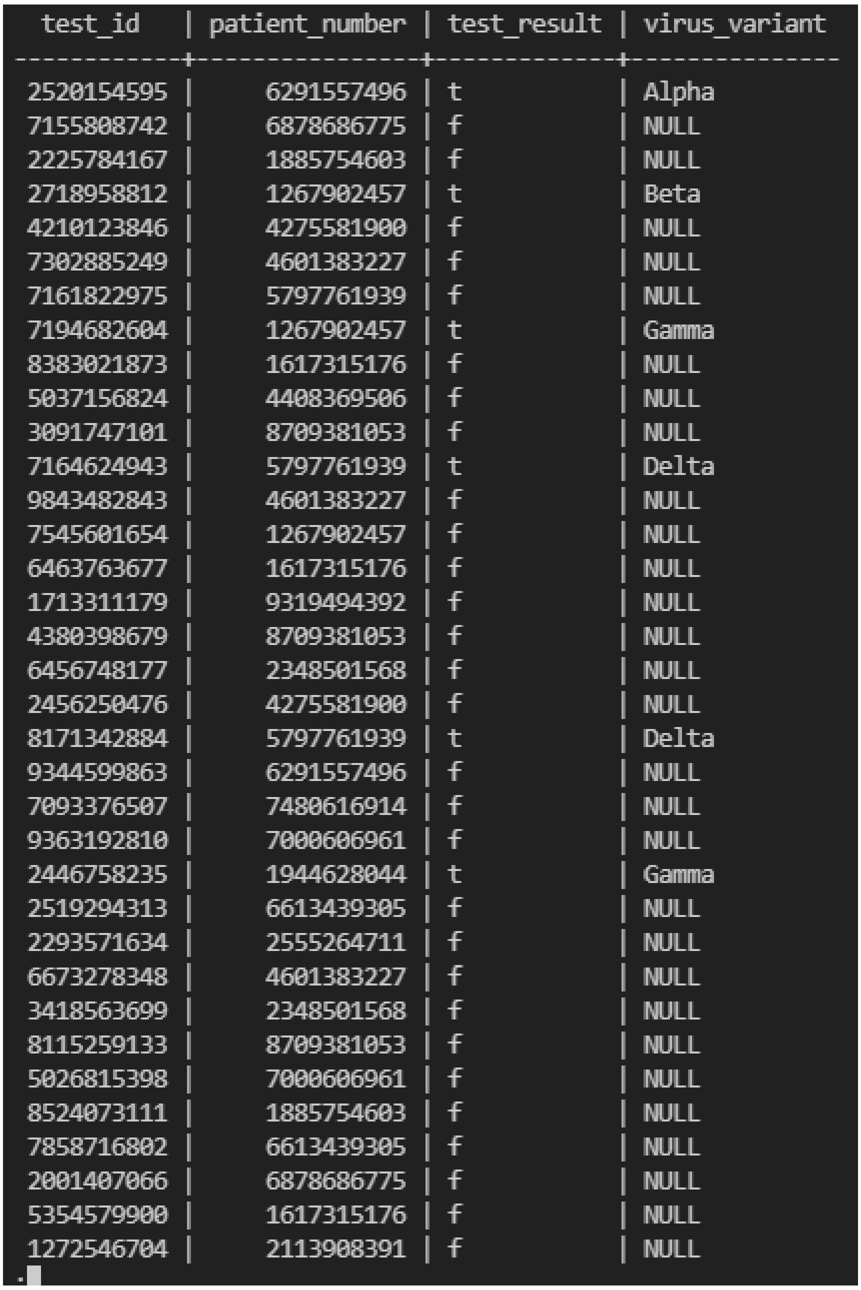
Test Record:

Description:



```
assignment2=# \d test_record
                     Table "public.test_record"
     Column     |         Type         | Collation | Nullable | Default
----------------+----------------------+-----------+----------+---------
 test_id        | bigint               |           | not null |
 patient_number | bigint               |           | not null |
 test_result    | boolean              |           | not null |
 virus_variant  | character varying(5) |           | not null |
Indexes:
    "test_record_test_id_key" UNIQUE CONSTRAINT, btree (test_id)
Foreign-key constraints:
    "test_record_patient_number_fkey" FOREIGN KEY (patient_number) REFERENCES patient(nhs_number)
    "test_record_test_id_fkey" FOREIGN KEY (test_id) REFERENCES test(test_id)
```

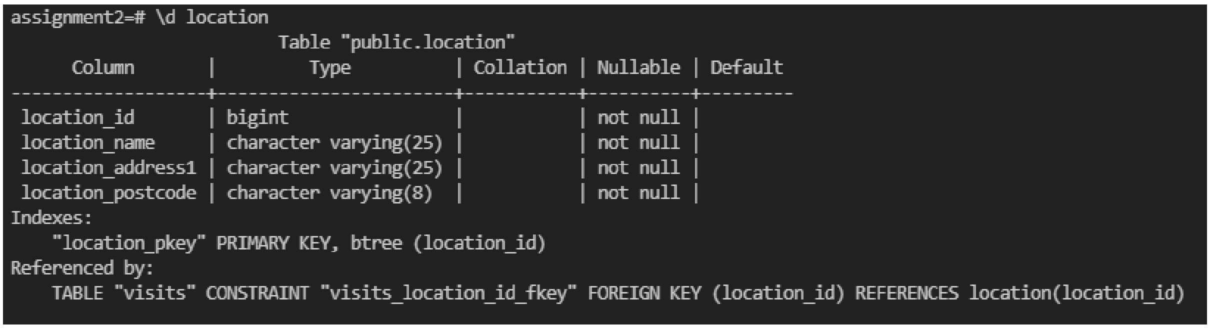
Dummy Data (Partial):



```
  test_id   | patient_number | test_result | virus_variant
------------+----------------+-------------+---------------
 2520154595 |     6291557496 | t           | Alpha
 7155808742 |     6878686775 | f           | NULL
 2225784167 |     1885754603 | f           | NULL
 2718958812 |     1267902457 | t           | Beta
 4210123846 |     4275581900 | f           | NULL
 7302885249 |     4601383227 | f           | NULL
 7161822975 |     5797761939 | f           | NULL
 7194682604 |     1267902457 | t           | Gamma
 8383021873 |     1617315176 | f           | NULL
 5037156824 |     4408369506 | f           | NULL
 3091747101 |     8709381053 | f           | NULL
 7164624943 |     5797761939 | t           | Delta
 9843482843 |     4601383227 | f           | NULL
 7545601654 |     1267902457 | f           | NULL
 6463763677 |     1617315176 | f           | NULL
 1713311179 |     9319494392 | f           | NULL
 4380398679 |     8709381053 | f           | NULL
 6456748177 |     2348501568 | f           | NULL
 2456250476 |     4275581900 | f           | NULL
 8171342884 |     5797761939 | t           | Delta
 9344599863 |     6291557496 | f           | NULL
 7093376507 |     7480616914 | f           | NULL
 9363192810 |     7000606961 | f           | NULL
 2446758235 |     1944628044 | t           | Gamma
 2519294313 |     6613439305 | f           | NULL
 2293571634 |     2555264711 | f           | NULL
 6673278348 |     4601383227 | f           | NULL
 3418563699 |     2348501568 | f           | NULL
 8115259133 |     8709381053 | f           | NULL
 5026815398 |     7000606961 | f           | NULL
 8524073111 |     1885754603 | f           | NULL
 7858716802 |     6613439305 | f           | NULL
 2001407066 |     6878686775 | f           | NULL
 5354579900 |     1617315176 | f           | NULL
 1272546704 |     2113908391 | f           | NULL
.
```

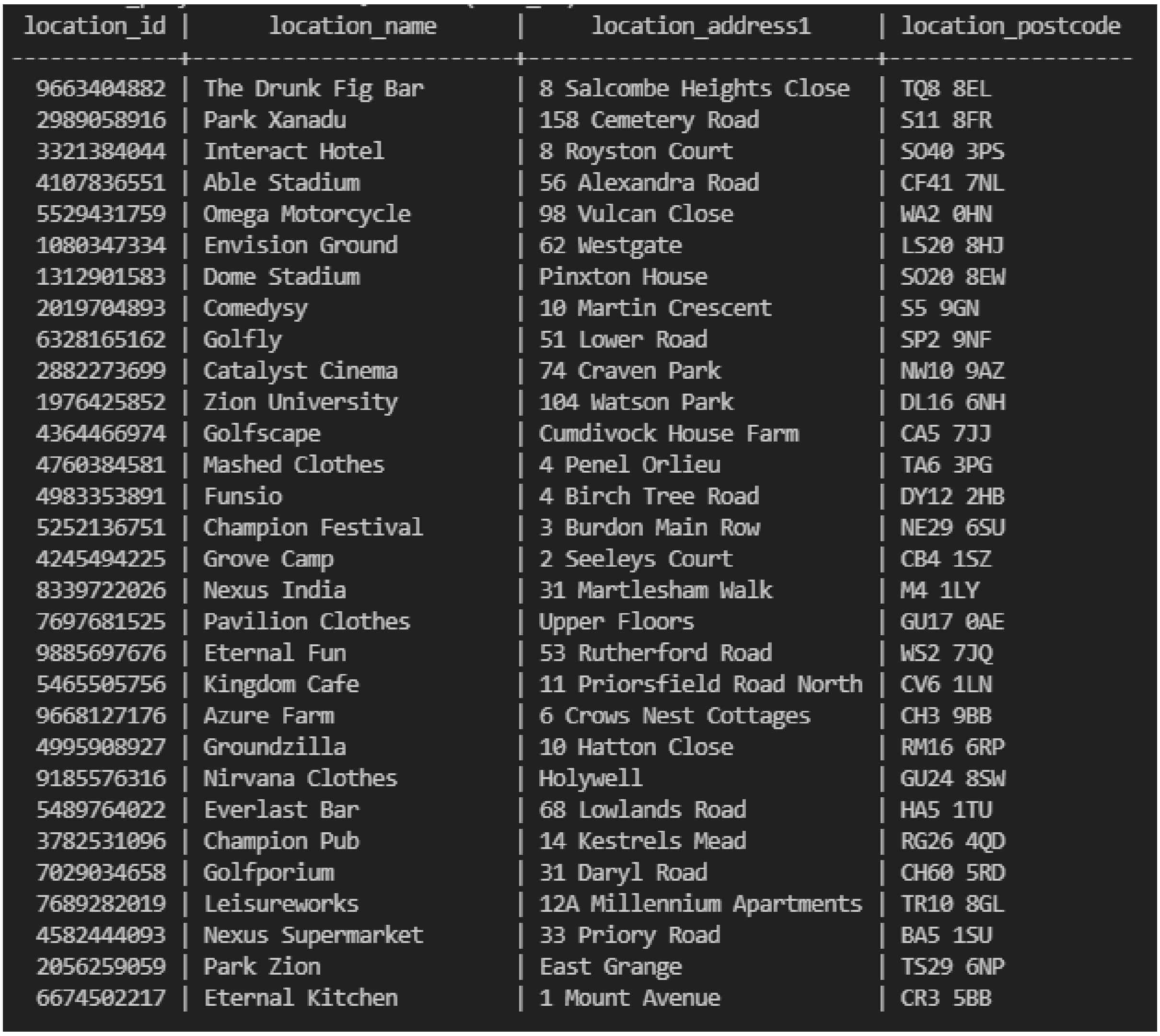
Location:

Description:



```
assignment2=# \d location
                        Table "public.location"
      Column       |         Type          | Collation | Nullable | Default
-------------------+-----------------------+-----------+----------+---------
 location_id       | bigint                |           | not null |
 location_name     | character varying(25) |           | not null |
 location_address1 | character varying(25) |           | not null |
 location_postcode | character varying(8)  |           | not null |
Indexes:
    "location_pkey" PRIMARY KEY, btree (location_id)
Referenced by:
    TABLE "visits" CONSTRAINT "visits_location_id_fkey" FOREIGN KEY (location_id) REFERENCES location(location_id)
```

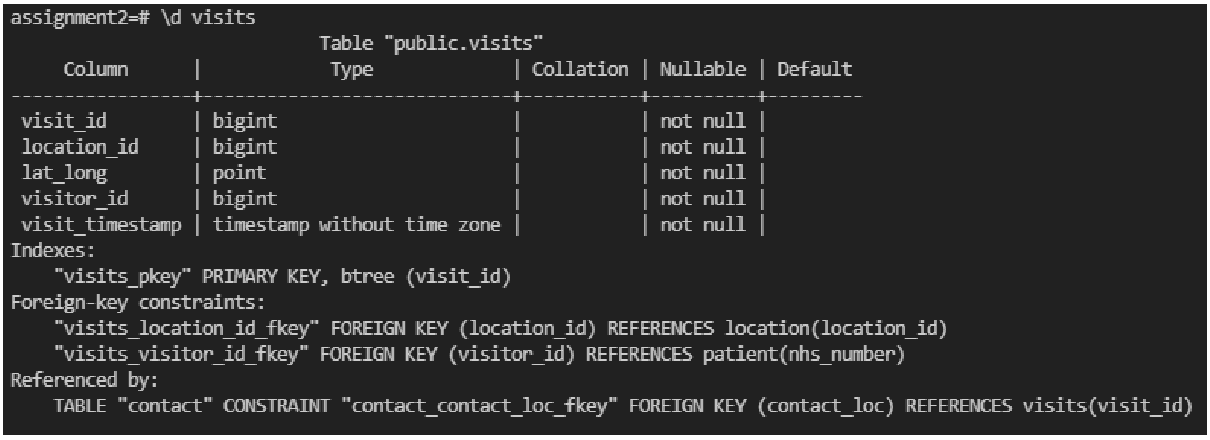
Dummy Data (Partial):



```
 location_id |      location_name       |     location_address1      | location_postcode
-------------+--------------------------+----------------------------+-------------------
  9663404882 | The Drunk Fig Bar        | 8 Salcombe Heights Close   | TQ8 8EL
  2989058916 | Park Xanadu              | 158 Cemetery Road          | S11 8FR
  3321384044 | Interact Hotel           | 8 Royston Court            | SO40 3PS
  4107836551 | Able Stadium             | 56 Alexandra Road          | CF41 7NL
  5529431759 | Omega Motorcycle         | 98 Vulcan Close            | WA2 0HN
  1080347334 | Envision Ground          | 62 Westgate                | LS20 8HJ
  1312901583 | Dome Stadium             | Pinxton House              | SO20 8EW
  2019704893 | Comedysy                 | 10 Martin Crescent         | S5 9GN
  6328165162 | Golfly                   | 51 Lower Road              | SP2 9NF
  2882273699 | Catalyst Cinema          | 74 Craven Park             | NW10 9AZ
  1976425852 | Zion University          | 104 Watson Park            | DL16 6NH
  4364466974 | Golfscape                | Cumdivock House Farm       | CA5 7JJ
  4760384581 | Mashed Clothes           | 4 Penel Orlieu             | TA6 3PG
  4983353891 | Funsio                   | 4 Birch Tree Road          | DY12 2HB
  5252136751 | Champion Festival        | 3 Burdon Main Row          | NE29 6SU
  4245494225 | Grove Camp               | 2 Seeleys Court            | CB4 1SZ
  8339722026 | Nexus India              | 31 Martlesham Walk         | M4 1LY
  7697681525 | Pavilion Clothes         | Upper Floors               | GU17 0AE
  9885697676 | Eternal Fun              | 53 Rutherford Road         | WS2 7JQ
  5465505756 | Kingdom Cafe             | 11 Priorsfield Road North  | CV6 1LN
  9668127176 | Azure Farm               | 6 Crows Nest Cottages      | CH3 9BB
  4995908927 | Groundzilla              | 10 Hatton Close            | RM16 6RP
  9185576316 | Nirvana Clothes          | Holywell                   | GU24 8SW
  5489764022 | Everlast Bar             | 68 Lowlands Road           | HA5 1TU
  3782531096 | Champion Pub             | 14 Kestrels Mead           | RG26 4QD
  7029034658 | Golfporium               | 31 Daryl Road              | CH60 5RD
  7689282019 | Leisureworks             | 12A Millennium Apartments  | TR10 8GL
  4582444093 | Nexus Supermarket        | 33 Priory Road             | BA5 1SU
  2056259059 | Park Zion                | East Grange                | TS29 6NP
  6674502217 | Eternal Kitchen          | 1 Mount Avenue             | CR3 5BB
```

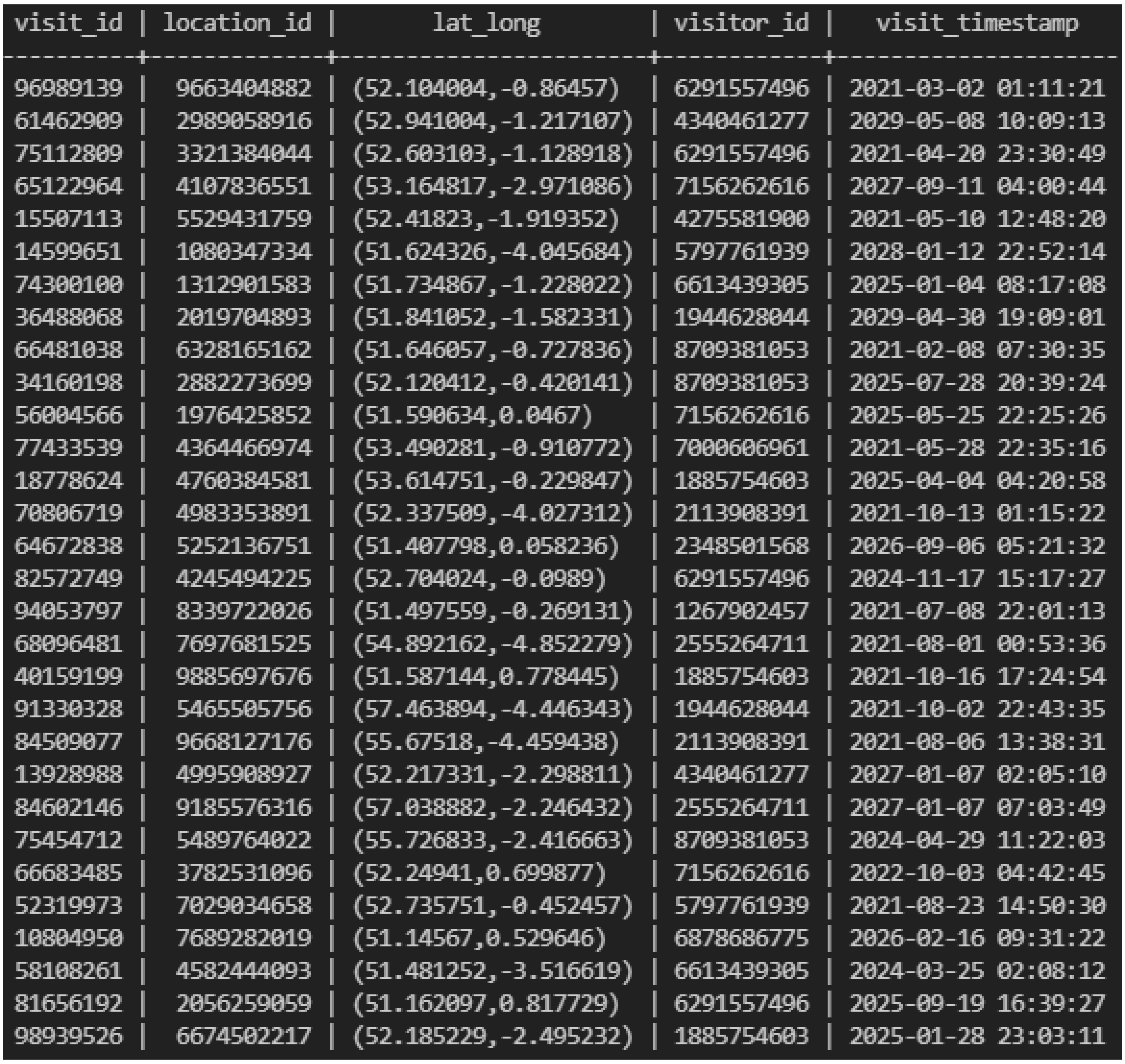
Visits:

Description:



```
assignment2=# \d visits
                          Table "public.visits"
     Column      |            Type             | Collation | Nullable | Default
-----------------+-----------------------------+-----------+----------+---------
 visit_id        | bigint                      |           | not null |
 location_id     | bigint                      |           | not null |
 lat_long        | point                       |           | not null |
 visitor_id      | bigint                      |           | not null |
 visit_timestamp | timestamp without time zone |           | not null |
Indexes:
    "visits_pkey" PRIMARY KEY, btree (visit_id)
Foreign-key constraints:
    "visits_location_id_fkey" FOREIGN KEY (location_id) REFERENCES location(location_id)
    "visits_visitor_id_fkey" FOREIGN KEY (visitor_id) REFERENCES patient(nhs_number)
Referenced by:
    TABLE "contact" CONSTRAINT "contact_contact_loc_fkey" FOREIGN KEY (contact_loc) REFERENCES visits(visit_id)
```

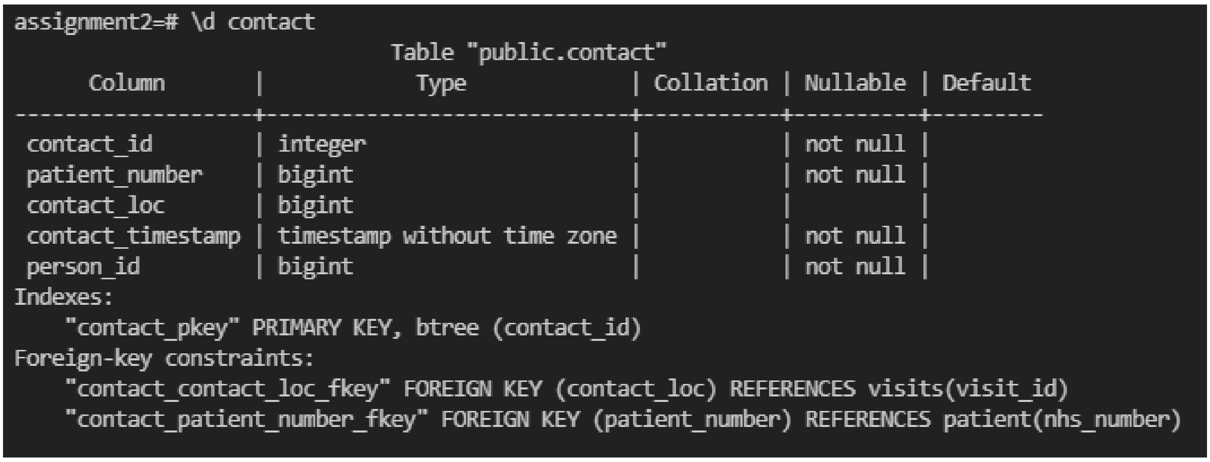
Dummy Data (Partial):



```
 visit_id | location_id |       lat_long        | visitor_id |   visit_timestamp
----------+-------------+-----------------------+------------+---------------------
 96989139 |  9663404882 | (52.104004,-0.86457)  | 6291557496 | 2021-03-02 01:11:21
 61462909 |  2989058916 | (52.941004,-1.217107) | 4340461277 | 2029-05-08 10:09:13
 75112809 |  3321384044 | (52.603103,-1.128918) | 6291557496 | 2021-04-20 23:30:49
 65122964 |  4107836551 | (53.164817,-2.971086) | 7156262616 | 2027-09-11 04:00:44
 15507113 |  5529431759 | (52.41823,-1.919352)  | 4275581900 | 2021-05-10 12:48:20
 14599651 |  1080347334 | (51.624326,-4.045684) | 5797761939 | 2028-01-12 22:52:14
 74300100 |  1312901583 | (51.734867,-1.228022) | 6613439305 | 2025-01-04 08:17:08
 36488068 |  2019704893 | (51.841052,-1.582331) | 1944628044 | 2029-04-30 19:09:01
 66481038 |  6328165162 | (51.646057,-0.727836) | 8709381053 | 2021-02-08 07:30:35
 34160198 |  2882273699 | (52.120412,-0.420141) | 8709381053 | 2025-07-28 20:39:24
 56004566 |  1976425852 | (51.590634,0.0467)    | 7156262616 | 2025-05-25 22:25:26
 77433539 |  4364466974 | (53.490281,-0.910772) | 7000606961 | 2021-05-28 22:35:16
 18778624 |  4760384581 | (53.614751,-0.229847) | 1885754603 | 2025-04-04 04:20:58
 70806719 |  4983353891 | (52.337509,-4.027312) | 2113908391 | 2021-10-13 01:15:22
 64672838 |  5252136751 | (51.407798,0.058236)  | 2348501568 | 2026-09-06 05:21:32
 82572749 |  4245494225 | (52.704024,-0.0989)   | 6291557496 | 2024-11-17 15:17:27
 94053797 |  8339722026 | (51.497559,-0.269131) | 1267902457 | 2021-07-08 22:01:13
 68096481 |  7697681525 | (54.892162,-4.852279) | 2555264711 | 2021-08-01 00:53:36
 40159199 |  9885697676 | (51.587144,0.778445)  | 1885754603 | 2021-10-16 17:24:54
 91330328 |  5465505756 | (57.463894,-4.446343) | 1944628044 | 2021-10-02 22:43:35
 84509077 |  9668127176 | (55.67518,-4.459438)  | 2113908391 | 2021-08-06 13:38:31
 13928988 |  4995908927 | (52.217331,-2.298811) | 4340461277 | 2027-01-07 02:05:10
 84602146 |  9185576316 | (57.038882,-2.246432) | 2555264711 | 2027-01-07 07:03:49
 75454712 |  5489764022 | (55.726833,-2.416663) | 8709381053 | 2024-04-29 11:22:03
 66683485 |  3782531096 | (52.24941,0.699877)   | 7156262616 | 2022-10-03 04:42:45
 52319973 |  7029034658 | (52.735751,-0.452457) | 5797761939 | 2021-08-23 14:50:30
 10804950 |  7689282019 | (51.14567,0.529646)   | 6878686775 | 2026-02-16 09:31:22
 58108261 |  4582444093 | (51.481252,-3.516619) | 6613439305 | 2024-03-25 02:08:12
 81656192 |  2056259059 | (51.162097,0.817729)  | 6291557496 | 2025-09-19 16:39:27
 98939526 |  6674502217 | (52.185229,-2.495232) | 1885754603 | 2025-01-28 23:03:11
```

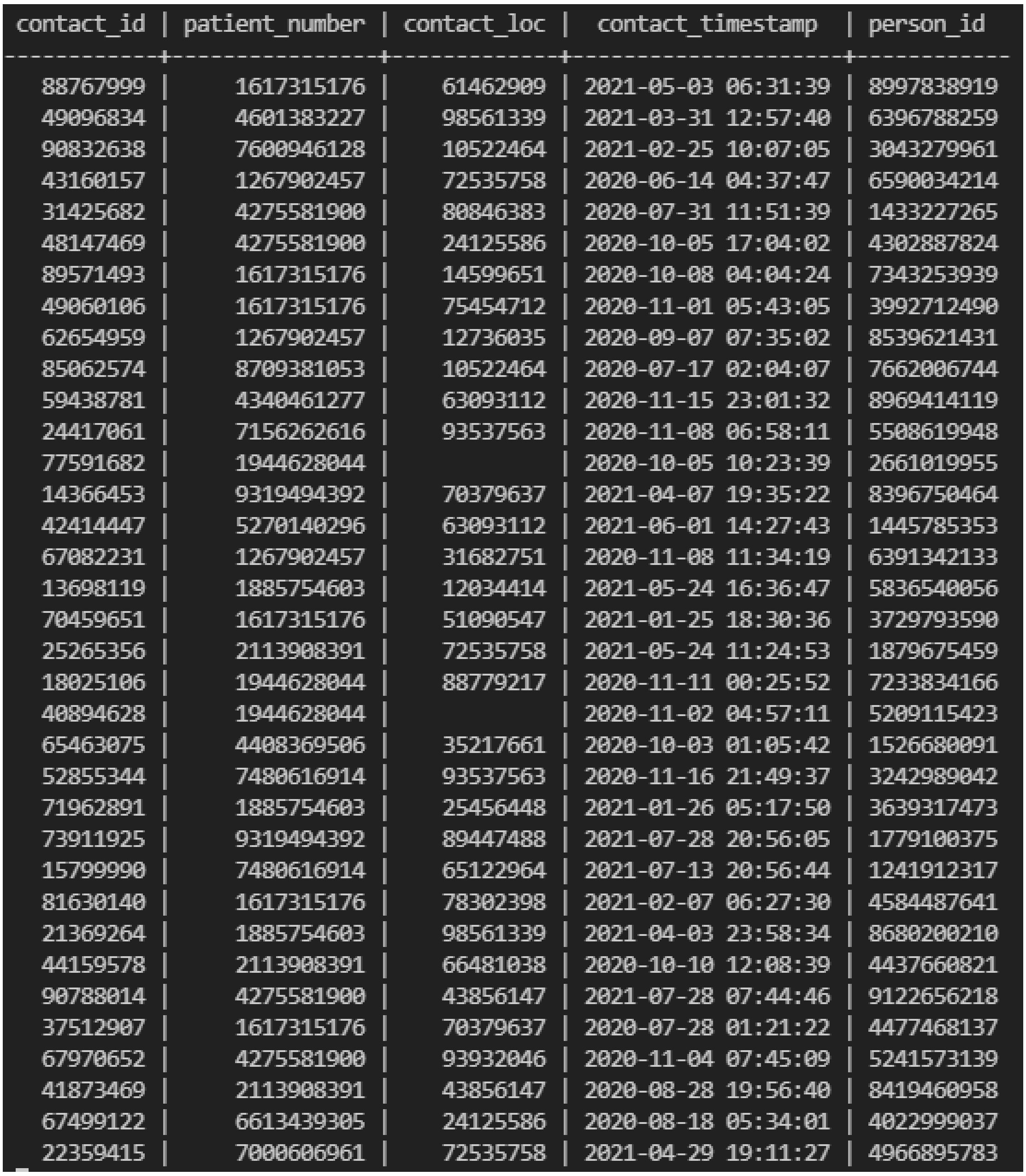
Contact:

Description:



```
assignment2=# \d contact
                          Table "public.contact"
      Column       |            Type             | Collation | Nullable | Default
-------------------+-----------------------------+-----------+----------+---------
 contact_id        | integer                     |           | not null |
 patient_number    | bigint                      |           | not null |
 contact_loc       | bigint                      |           |          |
 contact_timestamp | timestamp without time zone |           | not null |
 person_id         | bigint                      |           | not null |
Indexes:
    "contact_pkey" PRIMARY KEY, btree (contact_id)
Foreign-key constraints:
    "contact_contact_loc_fkey" FOREIGN KEY (contact_loc) REFERENCES visits(visit_id)
    "contact_patient_number_fkey" FOREIGN KEY (patient_number) REFERENCES patient(nhs_number)
```

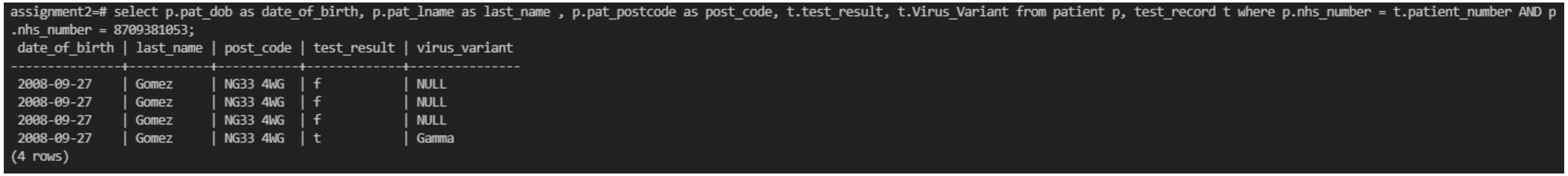
Dummy Data (Partial):



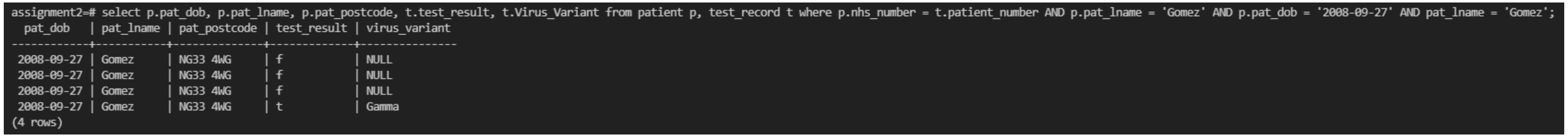
```
 contact_id | patient_number | contact_loc |  contact_timestamp  | person_id
------------+----------------+-------------+---------------------+------------
   88767999 |     1617315176 |    61462909 | 2021-05-03 06:31:39 | 8997838919
   49096834 |     4601383227 |    98561339 | 2021-03-31 12:57:40 | 6396788259
   90832638 |     7600946128 |    10522464 | 2021-02-25 10:07:05 | 3043279961
   43160157 |     1267902457 |    72535758 | 2020-06-14 04:37:47 | 6590034214
   31425682 |     4275581900 |    80846383 | 2020-07-31 11:51:39 | 1433227265
   48147469 |     4275581900 |    24125586 | 2020-10-05 17:04:02 | 4302887824
   89571493 |     1617315176 |    14599651 | 2020-10-08 04:04:24 | 7343253939
   49060106 |     1617315176 |    75454712 | 2020-11-01 05:43:05 | 3992712490
   62654959 |     1267902457 |    12736035 | 2020-09-07 07:35:02 | 8539621431
   85062574 |     8709381053 |    10522464 | 2020-07-17 02:04:07 | 7662006744
   59438781 |     4340461277 |    63093112 | 2020-11-15 23:01:32 | 8969414119
   24417061 |     7156262616 |    93537563 | 2020-11-08 06:58:11 | 5508619948
   77591682 |     1944628044 |             | 2020-10-05 10:23:39 | 2661019955
   14366453 |     9319494392 |    70379637 | 2021-04-07 19:35:22 | 8396750464
   42414447 |     5270140296 |    63093112 | 2021-06-01 14:27:43 | 1445785353
   67082231 |     1267902457 |    31682751 | 2020-11-08 11:34:19 | 6391342133
   13698119 |     1885754603 |    12034414 | 2021-05-24 16:36:47 | 5836540056
   70459651 |     1617315176 |    51090547 | 2021-01-25 18:30:36 | 3729793590
   25265356 |     2113908391 |    72535758 | 2021-05-24 11:24:53 | 1879675459
   18025106 |     1944628044 |    88779217 | 2020-11-11 00:25:52 | 7233834166
   40894628 |     1944628044 |             | 2020-11-02 04:57:11 | 5209115423
   65463075 |     4408369506 |    35217661 | 2020-10-03 01:05:42 | 1526680091
   52855344 |     7480616914 |    93537563 | 2020-11-16 21:49:37 | 3242989042
   71962891 |     1885754603 |    25456448 | 2021-01-26 05:17:50 | 3639317473
   73911925 |     9319494392 |    89447488 | 2021-07-28 20:56:05 | 1779100375
   15799990 |     7480616914 |    65122964 | 2021-07-13 20:56:44 | 1241912317
   81630140 |     1617315176 |    78302398 | 2021-02-07 06:27:30 | 4584487641
   21369264 |     1885754603 |    98561339 | 2021-04-03 23:58:34 | 8680200210
   44159578 |     2113908391 |    66481038 | 2020-10-10 12:08:39 | 4437660821
   90788014 |     4275581900 |    43856147 | 2021-07-28 07:44:46 | 9122656218
   37512907 |     1617315176 |    70379637 | 2020-07-28 01:21:22 | 4477468137
   67970652 |     4275581900 |    93932046 | 2020-11-04 07:45:09 | 5241573139
   41873469 |     2113908391 |    43856147 | 2020-08-28 19:56:40 | 8419460958
   67499122 |     6613439305 |    24125586 | 2020-08-18 05:34:01 | 4022999037
   22359415 |     7000606961 |    72535758 | 2021-04-29 19:11:27 | 4966895783
```

Task 3: Querying PostgreSQL database using SQL Queries

Test report: showing all the test result for a single participant identified by date of birth, family name and postcode

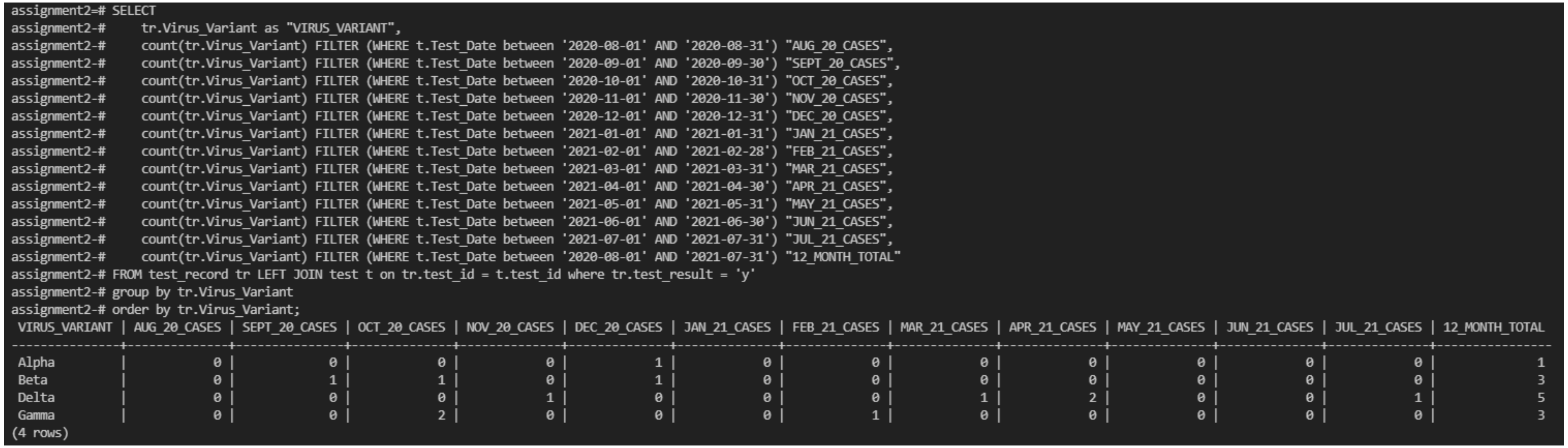


```
assignment2=# select p.pat_dob as date_of_birth, p.pat_lname as last_name , p.pat_postcode as post_code, t.test_result, t.Virus_Variant from patient p, test_record t where p.nhs_number = t.patient_number AND p
.nhs_number = 8709381053;
 date_of_birth | last_name | post_code | test_result | virus_variant
---------------+-----------+-----------+-------------+--------------
 2008-09-27    | Gomez     | NG33 4WG  | f           | NULL
 2008-09-27    | Gomez     | NG33 4WG  | f           | NULL
 2008-09-27    | Gomez     | NG33 4WG  | f           | NULL
 2008-09-27    | Gomez     | NG33 4WG  | t           | Gamma
(4 rows)
```



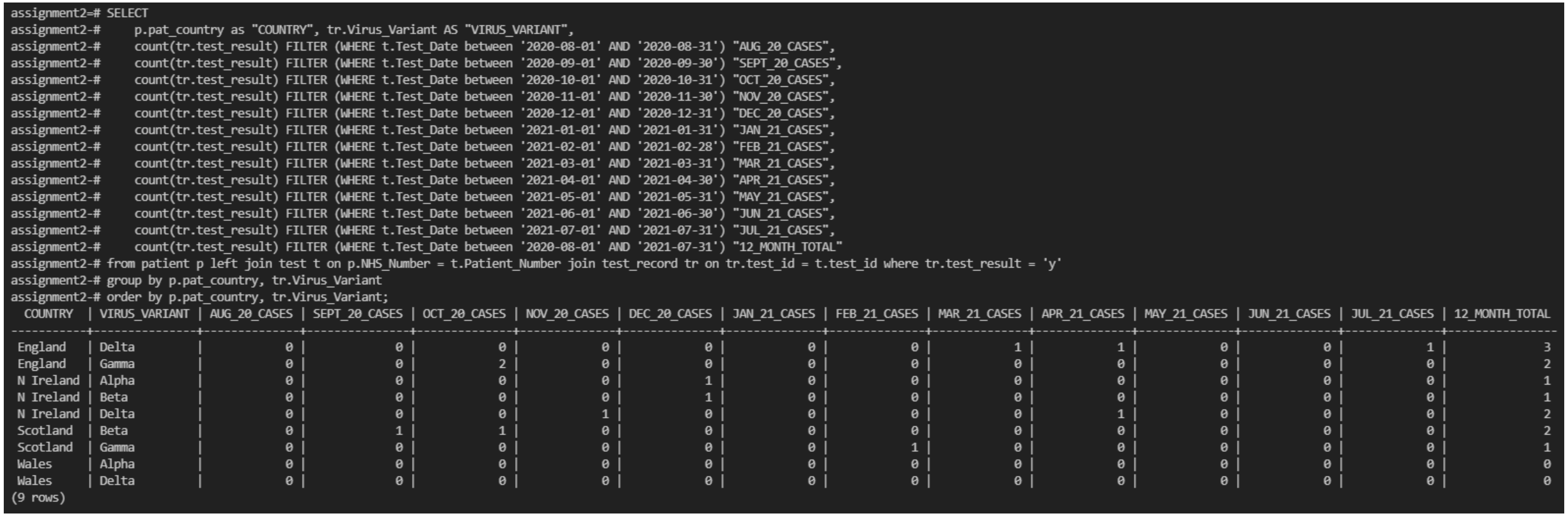
```
assignment2=# select p.pat_dob, p.pat_lname, p.pat_postcode, t.test_result, t.Virus_Variant from patient p, test_record t where p.nhs_number = t.patient_number AND p.pat_lname = 'Gomez' AND p.pat_dob = '2008-09-27' AND pat_lname = 'Gomez';
  pat_dob   | pat_lname | pat_postcode | test_result | virus_variant
------------+-----------+--------------+-------------+--------------
 2008-09-27 | Gomez     | NG33 4WG     | f           | NULL
 2008-09-27 | Gomez     | NG33 4WG     | f           | NULL
 2008-09-27 | Gomez     | NG33 4WG     | f           | NULL
 2008-09-27 | Gomez     | NG33 4WG     | t           | Gamma
(4 rows)
```

Virus variant distribution over the last 12 months



```
assignment2=# SELECT
assignment2-#     tr.Virus_Variant as "VIRUS_VARIANT",
assignment2-#     count(tr.Virus_Variant) FILTER (WHERE t.Test_Date between '2020-08-01' AND '2020-08-31') "AUG_20_CASES",
assignment2-#     count(tr.Virus_Variant) FILTER (WHERE t.Test_Date between '2020-09-01' AND '2020-09-30') "SEPT_20_CASES",
assignment2-#     count(tr.Virus_Variant) FILTER (WHERE t.Test_Date between '2020-10-01' AND '2020-10-31') "OCT_20_CASES",
assignment2-#     count(tr.Virus_Variant) FILTER (WHERE t.Test_Date between '2020-11-01' AND '2020-11-30') "NOV_20_CASES",
assignment2-#     count(tr.Virus_Variant) FILTER (WHERE t.Test_Date between '2020-12-01' AND '2020-12-31') "DEC_20_CASES",
assignment2-#     count(tr.Virus_Variant) FILTER (WHERE t.Test_Date between '2021-01-01' AND '2021-01-31') "JAN_21_CASES",
assignment2-#     count(tr.Virus_Variant) FILTER (WHERE t.Test_Date between '2021-02-01' AND '2021-02-28') "FEB_21_CASES",
assignment2-#     count(tr.Virus_Variant) FILTER (WHERE t.Test_Date between '2021-03-01' AND '2021-03-31') "MAR_21_CASES",
assignment2-#     count(tr.Virus_Variant) FILTER (WHERE t.Test_Date between '2021-04-01' AND '2021-04-30') "APR_21_CASES",
assignment2-#     count(tr.Virus_Variant) FILTER (WHERE t.Test_Date between '2021-05-01' AND '2021-05-31') "MAY_21_CASES",
assignment2-#     count(tr.Virus_Variant) FILTER (WHERE t.Test_Date between '2021-06-01' AND '2021-06-30') "JUN_21_CASES",
assignment2-#     count(tr.Virus_Variant) FILTER (WHERE t.Test_Date between '2021-07-01' AND '2021-07-31') "JUL_21_CASES",
assignment2-#     count(tr.Virus_Variant) FILTER (WHERE t.Test_Date between '2020-08-01' AND '2021-07-31') "12_MONTH_TOTAL"
assignment2-# FROM test_record tr LEFT JOIN test t on tr.test_id = t.test_id where tr.test_result = 'y'
assignment2-# group by tr.Virus_Variant
assignment2-# order by tr.Virus_Variant;
 VIRUS_VARIANT | AUG_20_CASES | SEPT_20_CASES | OCT_20_CASES | NOV_20_CASES | DEC_20_CASES | JAN_21_CASES | FEB_21_CASES | MAR_21_CASES | APR_21_CASES | MAY_21_CASES | JUN_21_CASES | JUL_21_CASES | 12_MONTH_TOTAL
---------------+--------------+---------------+--------------+--------------+--------------+--------------+--------------+--------------+--------------+--------------+--------------+--------------+---------------
 Alpha         |            0 |             0 |            0 |            0 |            1 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |              1
 Beta          |            0 |             1 |            1 |            0 |            1 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |              3
 Delta         |            0 |             0 |            0 |            1 |            0 |            0 |            0 |            1 |            2 |            0 |            0 |            1 |              5
 Gamma         |            0 |             0 |            2 |            0 |            0 |            0 |            1 |            0 |            0 |            0 |            0 |            0 |              3
(4 rows)
```

Number of positive tests over the last 12 months per country i.e. England, Scotland, Wales and Northern Ireland.



```
assignment2=# SELECT
assignment2-#     p.pat_country as "COUNTRY", tr.Virus_Variant AS "VIRUS_VARIANT",
assignment2-#     count(tr.test_result) FILTER (WHERE t.Test_Date between '2020-08-01' AND '2020-08-31') "AUG_20_CASES",
assignment2-#     count(tr.test_result) FILTER (WHERE t.Test_Date between '2020-09-01' AND '2020-09-30') "SEPT_20_CASES",
assignment2-#     count(tr.test_result) FILTER (WHERE t.Test_Date between '2020-10-01' AND '2020-10-31') "OCT_20_CASES",
assignment2-#     count(tr.test_result) FILTER (WHERE t.Test_Date between '2020-11-01' AND '2020-11-30') "NOV_20_CASES",
assignment2-#     count(tr.test_result) FILTER (WHERE t.Test_Date between '2020-12-01' AND '2020-12-31') "DEC_20_CASES",
assignment2-#     count(tr.test_result) FILTER (WHERE t.Test_Date between '2021-01-01' AND '2021-01-31') "JAN_21_CASES",
assignment2-#     count(tr.test_result) FILTER (WHERE t.Test_Date between '2021-02-01' AND '2021-02-28') "FEB_21_CASES",
assignment2-#     count(tr.test_result) FILTER (WHERE t.Test_Date between '2021-03-01' AND '2021-03-31') "MAR_21_CASES",
assignment2-#     count(tr.test_result) FILTER (WHERE t.Test_Date between '2021-04-01' AND '2021-04-30') "APR_21_CASES",
assignment2-#     count(tr.test_result) FILTER (WHERE t.Test_Date between '2021-05-01' AND '2021-05-31') "MAY_21_CASES",
assignment2-#     count(tr.test_result) FILTER (WHERE t.Test_Date between '2021-06-01' AND '2021-06-30') "JUN_21_CASES",
assignment2-#     count(tr.test_result) FILTER (WHERE t.Test_Date between '2021-07-01' AND '2021-07-31') "JUL_21_CASES",
assignment2-#     count(tr.test_result) FILTER (WHERE t.Test_Date between '2020-08-01' AND '2021-07-31') "12_MONTH_TOTAL"
assignment2-# from patient p left join test t on p.NHS_Number = t.Patient_Number join test_record tr on tr.test_id = t.test_id where tr.test_result = 'y'
assignment2-# group by p.pat_country, tr.Virus_Variant
assignment2-# order by p.pat_country, tr.Virus_Variant;
  COUNTRY  | VIRUS_VARIANT | AUG_20_CASES | SEPT_20_CASES | OCT_20_CASES | NOV_20_CASES | DEC_20_CASES | JAN_21_CASES | FEB_21_CASES | MAR_21_CASES | APR_21_CASES | MAY_21_CASES | JUN_21_CASES | JUL_21_CASES | 12_MONTH_TOTAL
-----------+---------------+--------------+---------------+--------------+--------------+--------------+--------------+--------------+--------------+--------------+--------------+--------------+--------------+----------------
 England   | Delta         |            0 |             0 |            0 |            0 |            0 |            0 |            0 |            1 |            1 |            0 |            0 |            1 |              3
 England   | Gamma         |            0 |             0 |            2 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |              2
 N Ireland | Alpha         |            0 |             0 |            0 |            0 |            1 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |              1
 N Ireland | Beta          |            0 |             0 |            0 |            0 |            1 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |              1
 N Ireland | Delta         |            0 |             0 |            0 |            1 |            0 |            0 |            0 |            0 |            1 |            0 |            0 |            0 |              2
 Scotland  | Beta          |            0 |             1 |            1 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |              2
 Scotland  | Gamma         |            0 |             0 |            0 |            0 |            0 |            0 |            1 |            0 |            0 |            0 |            0 |            0 |              1
 Wales     | Alpha         |            0 |             0 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |              0
 Wales     | Delta         |            0 |             0 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |            0 |              0
(9 rows)
```

Task 4: SQL vs NoSQL Comparison Report

The purpose of this report is to help inform decision making regarding the transition of NHS database facilities from PostgreSQL towards a yet to be determined NoSQL based DBMS. However, before continuing, it may be helpful to outline the differences between an SQL vs NoSQL DBMS, as outlined in table 1.

SQL		NoSQL
Based on the creation and adherence to a tabular data structure.	**Model**	Used for non-relational data, storing data as key-value pairs, document trees or in a graph structure
Tables are created linked through the use of primary and foreign keys.	**Data**	Key-value pairs create a hash table which is used to indicate the server storage location. It can also serve as a document data store, where values are linked to nested objects, e.g. XML, JSON or YAML files.
Strict schema outlining what kind of data can be stored in a column.	**Flexibility**	The values do not have a fixed schema and can be anything from primitive values to compound structures
Indexes optimised for the use of ACID transactions (Atomicity, Consistency, Isolation & Durability)	**Transactions**	Designed for CRUD operations (Create, Read, Update & Delete)
Strong	**Consistency**	Mirrors provide redundancy, but new data needs to be copied across all mirrors of a single partition—little consistency until mirroring operation is completed.
Restricted availability. Data on all nodes must match before further requests can be responded to.	**Availability**	All requests get a non-error response, regardless of whether the same key has differing value pairs on different mirrors.
Scale vertically. More hardware needs to be added (faster CPU, more storage) to handle increased load requirements.	**Scale**	Scale horizontally, possible to respond to increased demand by simply adding more machines.

Table 1: Comparison of SQL and NoSQL databases. (Gupta, 2021)

Current NoSQL Availability

NoSQL DBMS fall into categories based on their internal architecture and data delivery processes. These are key-value, column family, document stores and graph, and can be cloud-hosted (AWS, Azure or Google Cloud) or cloud-agnostic (Meier & Kaufmann, 2019). However, to provide a helpful overview, this report will outline four of the most popular NoSQL DBMS: MongoDB, Cassandra, Amazon DynamoDB & HBase.

MongoDB – a document-based architecture, presenting documents in a schema-less fashion, with data stored as JSON (or JSON-like) files (MongoDB, 2020b). Advantages of this storage method are significantly faster queries, reducing the computing power needed to serve responses to users. Data is replicated across multiple nodes, promoting redundancy in the event of local hardware failure. Disadvantages of MongoDB are found in the fact that management operations such as updating need to be completed manually by DBAs and are a time-consuming process. Furthermore, MongoDB requires the active set to fit on system RAM and therefore is associated with increased cost as database size increases (Solarwinds, 2020).

Cassandra – an open-source, wide-column based architecture, made prominent by its ability to handle large volumes of data while providing close to real-time analysis through its high availability. It utilises a column and row architecture similar to traditional RDBMS and utilises Cassandra Query Language, similar in syntax to traditional SQL and is easy for users with SQL experience to transition. Cassandra also offers a distributed ring architecture with multiple nodes offering quicker response times and recovery from failure. (*Apache*, 2020) However, this distributed architecture also means that data consistency can represent a problem when recovering from a significant hardware failure. Furthermore, the coordinator note can easily be burdened during the recovery period and refuse to respond to any other incoming requests potentially resulting in data loss (Sarna, 2018).

Amazon DynamoDB – a key-value pair based deployment offered solely through Amazon Web Services (AWS). This AWS based deployment means that different tables can be stored in different locations, meaning that disk read/write resources are wholly dedicated to each of those tables, resulting in far faster execution times. Furthermore, AWS facilitates automatic scaling of operation, allowing for increased available resources during peak traffic periods. However, policies relating auto-scaling mean that when high usage exceeds five minutes, queries will error, leading to data loss.

HBase – similarly to Cassandra, HBase is an open-source, wide-column distributed database founded on the architecture of Google's BigTable. It utilises the Hadoop Distributed File System (HDFS), which facilitates the storage of enormous datasets while providing quick responses to analytical queries. HBase can also be hosted across off the shelf server-grade hardware, meaning that it is cost-effective to deploy, particularly as storage requirements extend towards the petabyte scale. While HBase exceeds Cassandra in terms of immediate data consistency, its utilisation of a master-slave architecture also presents a single point of failure, meaning system recovery after a hardware failure can result in performance problems. HBase also does no have a query language built-in, with additional technologies needed to interrogate the data, placing additional compute requirements on associated hardware (IBM, 2020).

Comparing PostgreSQL to MongoDB

PostgreSQL operates as an object-relational database management system (RDBMS) using tables and associated schema to present data. A table consists of rows describing individual records with the same set of columns described. Primary keys are used to identify each row uniquely, while foreign keys ensure referential integrity between tables. MongoDB, on the other hand, is a document store based DBMS, with documents in MongoDB analogous to PostgreSQL's records. MongoDB presents a schema-free DBMS, meaning that documents can be added, deleted or modified within the file structure without first creating a structure for that document.

PostgreSQL is compliant with American National Standards Institute (ANSI) 1999 Standardised SQL and partially compliant with ANSI:2003 (PostgreSQL Foundation, 2013). As such, familiarity with SQL will allow any user to execute DDL and DML operations without the need for additional documentation. MongoDB does not utilise SQL as a result of its JSON based document structure; instead, utilising a JSON query syntax differing substantially from standard SQL. A comparison of query syntax can be seen in table 2.

Query Type	PostgreSQL Query	MongoDB Query
Selecting records from the patient table	SELECT * from patient;	db.patient.find()
Inserting records into the patient table	INSERT INTO patient(NHS_Number, Pat_Fname, Pat_Lname) VALUES (001, Jane, Bloggs);	db.patient.insert({ NHS_Number: '001', Pat_Fname: 'Jane', Pat_Lname: 'Bloggs' })
Updating records in patient table.	UPDATE patient SET Test_Outcome= 'Positive' WHERE NHS_Number = 001;	db.patient.update({NHS_Number: '001'}, {$set: {Test_Outcome: 'Positive'}}, {multi: false})

Table 2: Comparison of PostgreSQL and MongoDB query syntax (Bui, 2021).

Both structures use a form of replication to ensure data redundancy with the aim of avoiding data loss in the case of hardware failure. PostgreSQL operates two synchronised database instances, one master one slave, with updates written to both instances mirroring one another after update operations are completed. This means that if the master instance fails, the slave instance will hold a full backup of the database ready for use and eventual restoration of the master. MongoDB achieves redundancy through files written to the primary node, with ideally two secondary nodes replicating the primary nodes operations log to ensure complete redundancy (Figure 1). In the event of a failure of the primary node hardware, eligible secondary nodes will elect to select a suitable new primary node, facilitating a quick handoff of database operations.

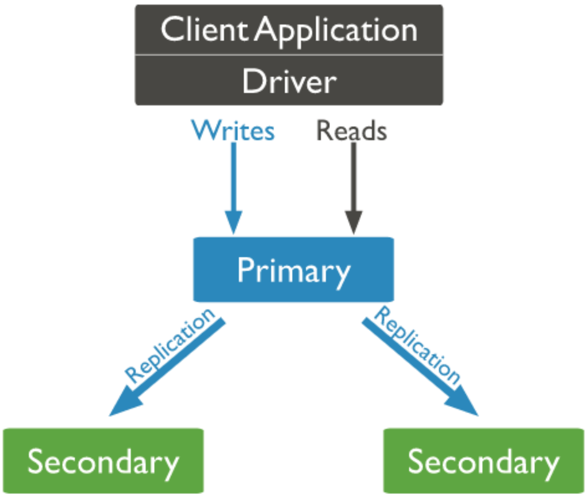


Figure 1: Replication in MongoDB (MongoDB 2020a).

It is important to note that MongoDB's primary focus is document-based data. As such, the expectation would be for it to out-perform PostgreSQL in JSON-based online analytical processing tasks. However, EDB (2019) illustrates that when using the industry-standard sysbench benchmarking tool, PostgreSQL performed three times faster than MongoDB on average across various workloads. Arguably this performance advantage is based on the maturity of PostgreSQL, and the years of community driven development work that has facilitated its performance, allowing it to compete on an even plane with NoSQL technologies like MongoDB.

In conclusion, both DBMS providers offer similar services, focused on different workflows and use cases. MongoDB provides scalability and realtime analytics and can be fundamental in content management, IOT and mobile app based workflows; particularly suitable where no clear schmea definition can be reached. On the other hand, PostgreSQL remains a traditional, resource-efficient and transactionally focused DBMS, particularly helpful in financial, geospatial and manufacturing focused workflows.

References

Apache. (2020). *Apache Cassandra Documentation*. Apache Cassandra. Retrieved 17 August 2021, from https://cassandra.apache.org/_/cassandra-basics.html

Arora, L. (2020, September 21). *5 Popular NoSQL Databases Every Data Science Professional Should Know About*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/09/different-nosql-databases-every-data-scientist-must-know/

Bui, A. (2021, June 2). *PostgreSQL vs. MongoDB*. Panoply. https://blog.panoply.io/postgresql-vs-mongodb

EDB. (2019). *New Benchmarks Show Postgres Dominating MongoDB in Varied Workloads*. https://www.enterprisedb.com/news/new-benchmarks-show-postgres-dominating-mongodb-varied-workloads

Gupta, S. C. (2021, August 2). *SQL vs. NoSQL Database: When to Use, How to Choose*. Medium. https://towardsdatascience.com/datastore-choices-sql-vs-nosql-database-ebec24d56106#ebbb

IBM. (2020). What is HBase? https://www.ibm.com/analytics/hadoop/hbase

Meier, A., & Kaufmann, M. (2019). *SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management* (1st ed. 2019 ed.) [E-book]. Springer Vieweg.

MongoDB. (2020a). Replication — MongoDB Manual. https://docs.mongodb.com/manual/replication/

MongoDB. (2020b, July 10). *Why Use MongoDB & When to Use It?*. MongoDB. https://www.mongodb.com/why-use-mongodb

PostgreSQL Foundation. (2013, February 7). SQL Conformance. PostgreSQL Documentation. https://www.postgresql.org/docs/8.3/features.html

Sarna, A. (2018, August 17). Is Apache Cassandra really the Database you need? Knoldus Blogs. https://blog.knoldus.com/is-apache-cassandra-really-the-database-you-need/

Solarwinds. (2020, March 13). *How to Tell if Your MongoDB Server Is Correctly Sized For Your Working Set*. Orange Matter. https://orangematter.solarwinds.com/2017/06/05/how-to-tell-if-your-mongodb-server-is-correctly-sized-for-your-working-set/

Appendices:

Appendix 1: Assignment Assumptions

1. Contacts created through handshake between phone app, this assumes mandated 100% usage of app by population.
2. Patients visit locations, but not all contacts necessarily occur at locations within database, so contact and location tables are treated sepeately with regards to the ERD.
3. Testing can be completed at home via a lateral flow test, meaning that not all tests will have an associated testing location
4. Patients are asked to check-in to locations where they have visited for >5 minutes, giving a location name where necessary.

Appendix 2: Full Database File

https://github.com/Brian-M-Collins/Data-Science-Portfolio/blob/main/NHS%20Track%20and%20Trace%20(.sql)/Database%20Code.sql