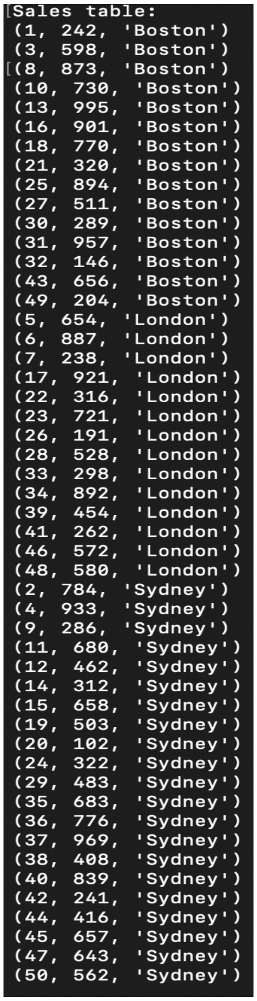
Distributed Database Systems
CSE 512 – Fall 2024

Assignment 1

Name: Sagar Rajesh Kumar Sinha **ASU ID**: 1229611034

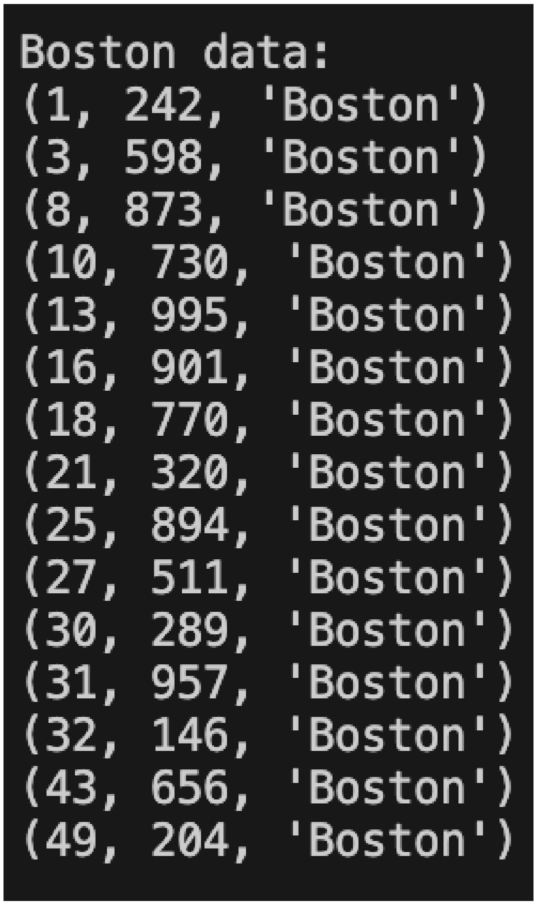
List Partitioning

List partitioning is a data organization method that splits a large dataset into smaller segments according to predefined attribute values. Unlike range partitioning, which uses continuous ranges, list partitioning uses a specific set of discrete values for partitioning. A designated partitioning key is assigned a list of these values, and any data entry that matches one of the values is placed in the corresponding partition. If no match is found, the entry is assigned to a different appropriate partition.

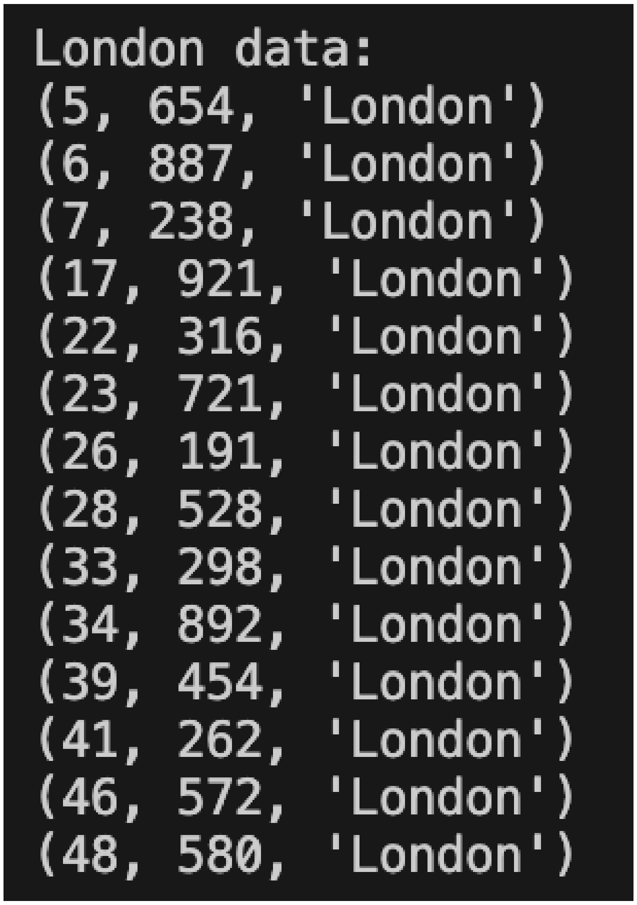


```
[Sales table:
 (1, 242, 'Boston')
 (3, 598, 'Boston')
[(8, 873, 'Boston')
 (10, 730, 'Boston')
 (13, 995, 'Boston')
 (16, 901, 'Boston')
 (18, 770, 'Boston')
 (21, 320, 'Boston')
 (25, 894, 'Boston')
 (27, 511, 'Boston')
 (30, 289, 'Boston')
 (31, 957, 'Boston')
 (32, 146, 'Boston')
 (43, 656, 'Boston')
 (49, 204, 'Boston')
 (5, 654, 'London')
 (6, 887, 'London')
 (7, 238, 'London')
 (17, 921, 'London')
 (22, 316, 'London')
 (23, 721, 'London')
 (26, 191, 'London')
 (28, 528, 'London')
 (33, 298, 'London')
 (34, 892, 'London')
 (39, 454, 'London')
 (41, 262, 'London')
 (46, 572, 'London')
 (48, 580, 'London')
 (2, 784, 'Sydney')
 (4, 933, 'Sydney')
 (9, 286, 'Sydney')
 (11, 680, 'Sydney')
 (12, 462, 'Sydney')
 (14, 312, 'Sydney')
 (15, 658, 'Sydney')
 (19, 503, 'Sydney')
 (20, 102, 'Sydney')
 (24, 322, 'Sydney')
 (29, 483, 'Sydney')
 (35, 683, 'Sydney')
 (36, 776, 'Sydney')
 (37, 969, 'Sydney')
 (38, 408, 'Sydney')
 (40, 839, 'Sydney')
 (42, 241, 'Sydney')
 (44, 416, 'Sydney')
 (45, 657, 'Sydney')
 (47, 643, 'Sydney')
 (50, 562, 'Sydney')
```

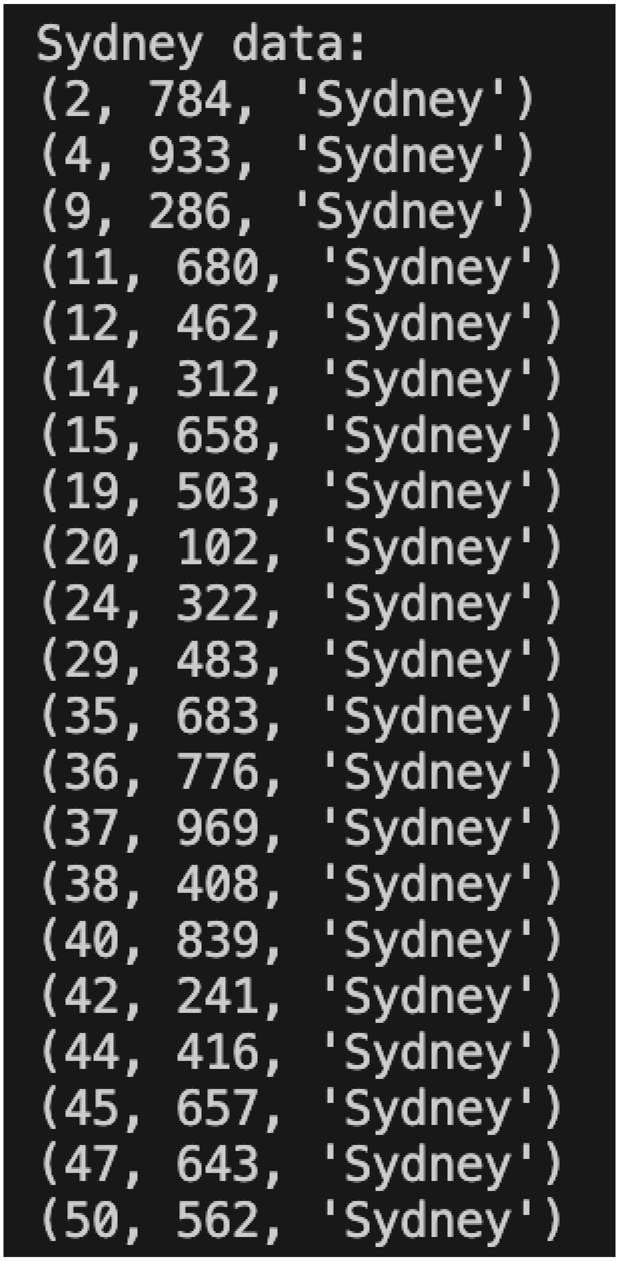
Assignment 1



```
Boston data:
(1, 242, 'Boston')
(3, 598, 'Boston')
(8, 873, 'Boston')
(10, 730, 'Boston')
(13, 995, 'Boston')
(16, 901, 'Boston')
(18, 770, 'Boston')
(21, 320, 'Boston')
(25, 894, 'Boston')
(27, 511, 'Boston')
(30, 289, 'Boston')
(31, 957, 'Boston')
(32, 146, 'Boston')
(43, 656, 'Boston')
(49, 204, 'Boston')
```



```
London data:
(5, 654, 'London')
(6, 887, 'London')
(7, 238, 'London')
(17, 921, 'London')
(22, 316, 'London')
(23, 721, 'London')
(26, 191, 'London')
(28, 528, 'London')
(33, 298, 'London')
(34, 892, 'London')
(39, 454, 'London')
(41, 262, 'London')
(46, 572, 'London')
(48, 580, 'London')
```



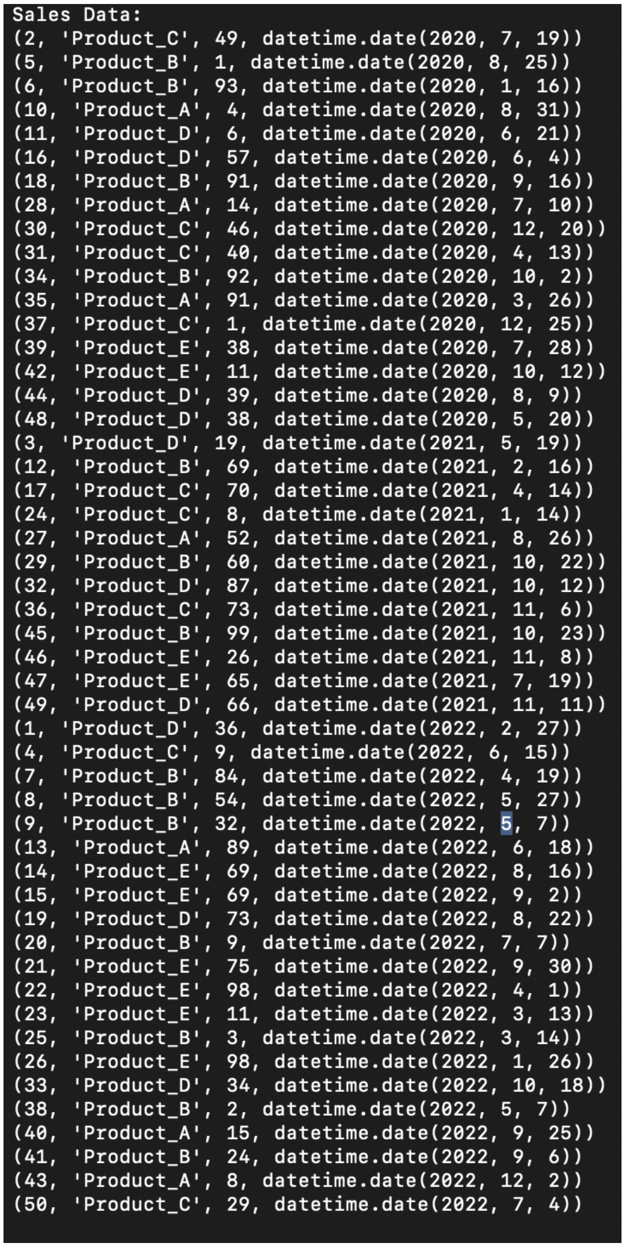
```
Sydney data:
(2, 784, 'Sydney')
(4, 933, 'Sydney')
(9, 286, 'Sydney')
(11, 680, 'Sydney')
(12, 462, 'Sydney')
(14, 312, 'Sydney')
(15, 658, 'Sydney')
(19, 503, 'Sydney')
(20, 102, 'Sydney')
(24, 322, 'Sydney')
(29, 483, 'Sydney')
(35, 683, 'Sydney')
(36, 776, 'Sydney')
(37, 969, 'Sydney')
(38, 408, 'Sydney')
(40, 839, 'Sydney')
(42, 241, 'Sydney')
(44, 416, 'Sydney')
(45, 657, 'Sydney')
(47, 643, 'Sydney')
(50, 562, 'Sydney')
```

Distributed Database Systems
CSE 512 – Fall 2024

Assignment 1

Range Partitioning

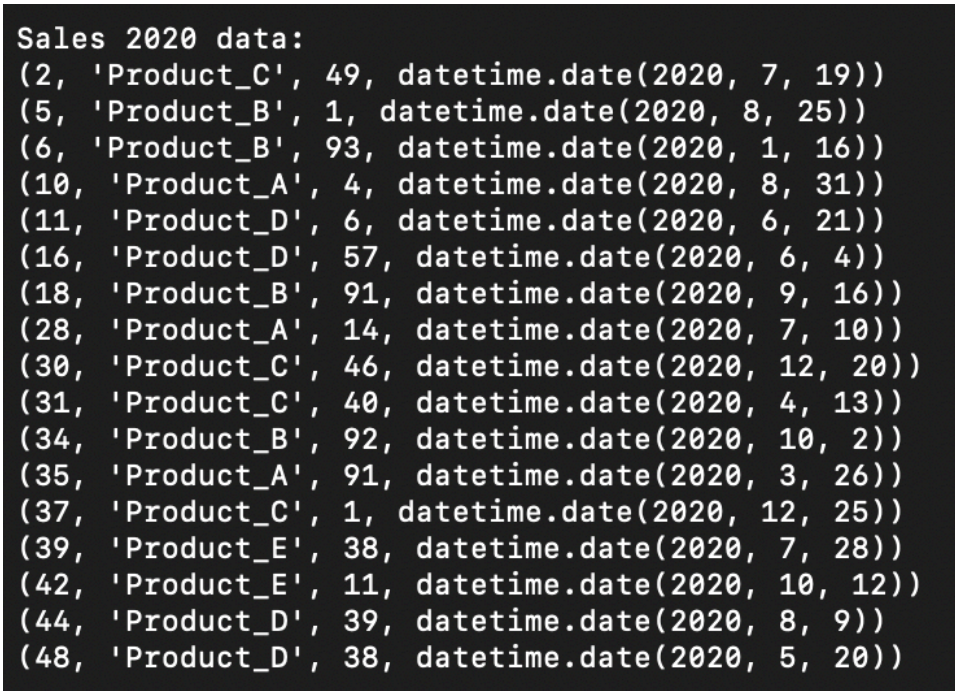
Range partitioning is a method used in relational databases where data is divided into partitions based on a set range for a particular field, such as IDs, dates, or numeric values like currency amounts. A specific range is associated with a partition key column, and when a data entry falls within that range, it is stored in the corresponding partition. If the entry doesn't match the range, it is directed to another partition that accommodates it.



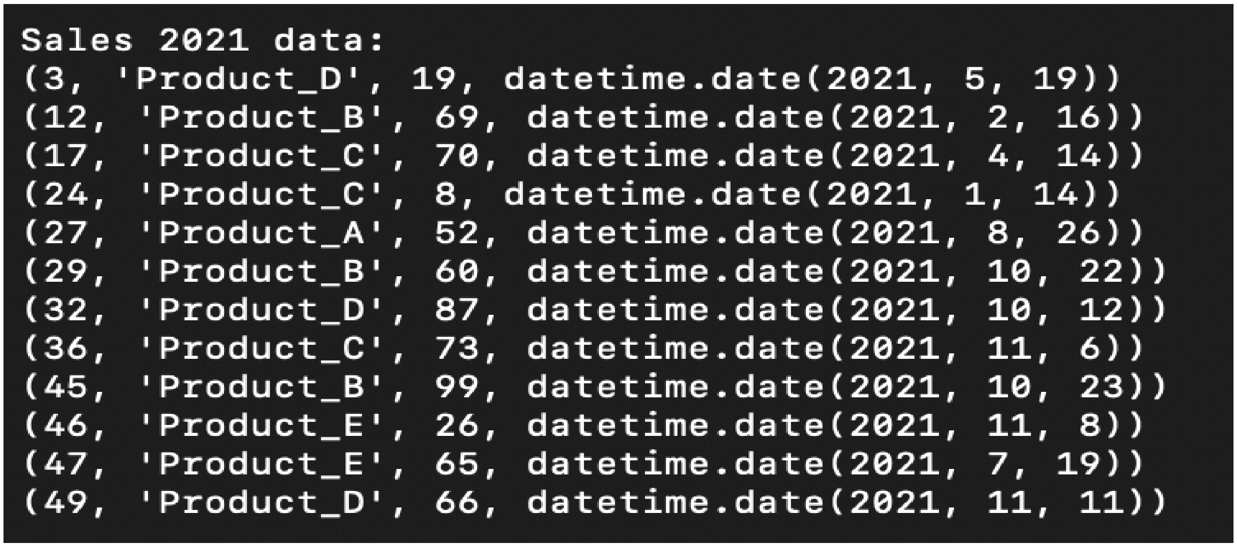
```
Sales Data:
(2, 'Product_C', 49, datetime.date(2020, 7, 19))
(5, 'Product_B', 1, datetime.date(2020, 8, 25))
(6, 'Product_B', 93, datetime.date(2020, 1, 16))
(10, 'Product_A', 4, datetime.date(2020, 8, 31))
(11, 'Product_D', 6, datetime.date(2020, 6, 21))
(16, 'Product_D', 57, datetime.date(2020, 6, 4))
(18, 'Product_B', 91, datetime.date(2020, 9, 16))
(28, 'Product_A', 14, datetime.date(2020, 7, 10))
(30, 'Product_C', 46, datetime.date(2020, 12, 20))
(31, 'Product_C', 40, datetime.date(2020, 4, 13))
(34, 'Product_B', 92, datetime.date(2020, 10, 2))
(35, 'Product_A', 91, datetime.date(2020, 3, 26))
(37, 'Product_C', 1, datetime.date(2020, 12, 25))
(39, 'Product_E', 38, datetime.date(2020, 7, 28))
(42, 'Product_E', 11, datetime.date(2020, 10, 12))
(44, 'Product_D', 39, datetime.date(2020, 8, 9))
(48, 'Product_D', 38, datetime.date(2020, 5, 20))
(3, 'Product_D', 19, datetime.date(2021, 5, 19))
(12, 'Product_B', 69, datetime.date(2021, 2, 16))
(17, 'Product_C', 70, datetime.date(2021, 4, 14))
(24, 'Product_C', 8, datetime.date(2021, 1, 14))
(27, 'Product_A', 52, datetime.date(2021, 8, 26))
(29, 'Product_B', 60, datetime.date(2021, 10, 22))
(32, 'Product_D', 87, datetime.date(2021, 10, 12))
(36, 'Product_C', 73, datetime.date(2021, 11, 6))
(45, 'Product_B', 99, datetime.date(2021, 10, 23))
(46, 'Product_E', 26, datetime.date(2021, 11, 8))
(47, 'Product_E', 65, datetime.date(2021, 7, 19))
(49, 'Product_D', 66, datetime.date(2021, 11, 11))
(1, 'Product_D', 36, datetime.date(2022, 2, 27))
(4, 'Product_C', 9, datetime.date(2022, 6, 15))
(7, 'Product_B', 84, datetime.date(2022, 4, 19))
(8, 'Product_B', 54, datetime.date(2022, 5, 27))
(9, 'Product_B', 32, datetime.date(2022, 5, 7))
(13, 'Product_A', 89, datetime.date(2022, 6, 18))
(14, 'Product_E', 69, datetime.date(2022, 8, 16))
(15, 'Product_E', 69, datetime.date(2022, 9, 2))
(19, 'Product_D', 73, datetime.date(2022, 8, 22))
(20, 'Product_B', 9, datetime.date(2022, 7, 7))
(21, 'Product_E', 75, datetime.date(2022, 9, 30))
(22, 'Product_E', 98, datetime.date(2022, 4, 1))
(23, 'Product_E', 11, datetime.date(2022, 3, 13))
(25, 'Product_B', 3, datetime.date(2022, 3, 14))
(26, 'Product_E', 98, datetime.date(2022, 1, 26))
(33, 'Product_D', 34, datetime.date(2022, 10, 18))
(38, 'Product_B', 2, datetime.date(2022, 5, 7))
(40, 'Product_A', 15, datetime.date(2022, 9, 25))
(41, 'Product_B', 24, datetime.date(2022, 9, 6))
(43, 'Product_A', 8, datetime.date(2022, 12, 2))
(50, 'Product_C', 29, datetime.date(2022, 7, 4))
```

Distributed Database Systems
CSE 512 – Fall 2024

Assignment 1



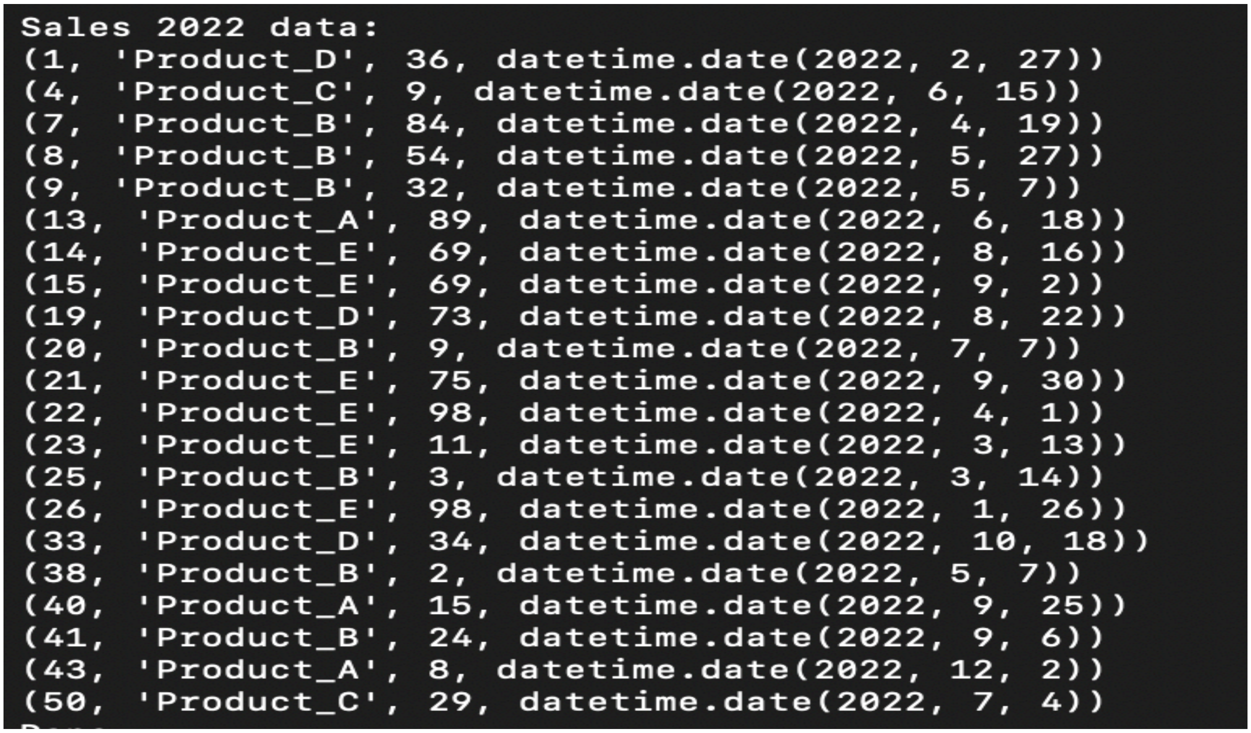
```
Sales 2020 data:
(2, 'Product_C', 49, datetime.date(2020, 7, 19))
(5, 'Product_B', 1, datetime.date(2020, 8, 25))
(6, 'Product_B', 93, datetime.date(2020, 1, 16))
(10, 'Product_A', 4, datetime.date(2020, 8, 31))
(11, 'Product_D', 6, datetime.date(2020, 6, 21))
(16, 'Product_D', 57, datetime.date(2020, 6, 4))
(18, 'Product_B', 91, datetime.date(2020, 9, 16))
(28, 'Product_A', 14, datetime.date(2020, 7, 10))
(30, 'Product_C', 46, datetime.date(2020, 12, 20))
(31, 'Product_C', 40, datetime.date(2020, 4, 13))
(34, 'Product_B', 92, datetime.date(2020, 10, 2))
(35, 'Product_A', 91, datetime.date(2020, 3, 26))
(37, 'Product_C', 1, datetime.date(2020, 12, 25))
(39, 'Product_E', 38, datetime.date(2020, 7, 28))
(42, 'Product_E', 11, datetime.date(2020, 10, 12))
(44, 'Product_D', 39, datetime.date(2020, 8, 9))
(48, 'Product_D', 38, datetime.date(2020, 5, 20))
```



```
Sales 2021 data:
(3, 'Product_D', 19, datetime.date(2021, 5, 19))
(12, 'Product_B', 69, datetime.date(2021, 2, 16))
(17, 'Product_C', 70, datetime.date(2021, 4, 14))
(24, 'Product_C', 8, datetime.date(2021, 1, 14))
(27, 'Product_A', 52, datetime.date(2021, 8, 26))
(29, 'Product_B', 60, datetime.date(2021, 10, 22))
(32, 'Product_D', 87, datetime.date(2021, 10, 12))
(36, 'Product_C', 73, datetime.date(2021, 11, 6))
(45, 'Product_B', 99, datetime.date(2021, 10, 23))
(46, 'Product_E', 26, datetime.date(2021, 11, 8))
(47, 'Product_E', 65, datetime.date(2021, 7, 19))
(49, 'Product_D', 66, datetime.date(2021, 11, 11))
```

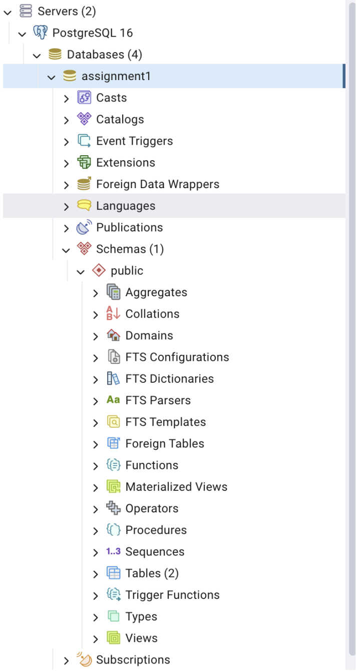
Distributed Database Systems
CSE 512 – Fall 2024

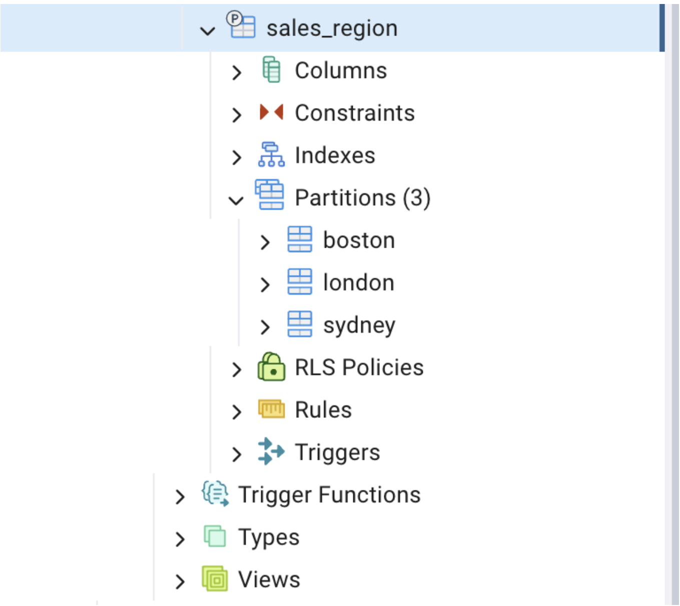
Assignment 1



```
Sales 2022 data:
(1, 'Product_D', 36, datetime.date(2022, 2, 27))
(4, 'Product_C', 9, datetime.date(2022, 6, 15))
(7, 'Product_B', 84, datetime.date(2022, 4, 19))
(8, 'Product_B', 54, datetime.date(2022, 5, 27))
(9, 'Product_B', 32, datetime.date(2022, 5, 7))
(13, 'Product_A', 89, datetime.date(2022, 6, 18))
(14, 'Product_E', 69, datetime.date(2022, 8, 16))
(15, 'Product_E', 69, datetime.date(2022, 9, 2))
(19, 'Product_D', 73, datetime.date(2022, 8, 22))
(20, 'Product_B', 9, datetime.date(2022, 7, 7))
(21, 'Product_E', 75, datetime.date(2022, 9, 30))
(22, 'Product_E', 98, datetime.date(2022, 4, 1))
(23, 'Product_E', 11, datetime.date(2022, 3, 13))
(25, 'Product_B', 3, datetime.date(2022, 3, 14))
(26, 'Product_E', 98, datetime.date(2022, 1, 26))
(33, 'Product_D', 34, datetime.date(2022, 10, 18))
(38, 'Product_B', 2, datetime.date(2022, 5, 7))
(40, 'Product_A', 15, datetime.date(2022, 9, 25))
(41, 'Product_B', 24, datetime.date(2022, 9, 6))
(43, 'Product_A', 8, datetime.date(2022, 12, 2))
(50, 'Product_C', 29, datetime.date(2022, 7, 4))
```

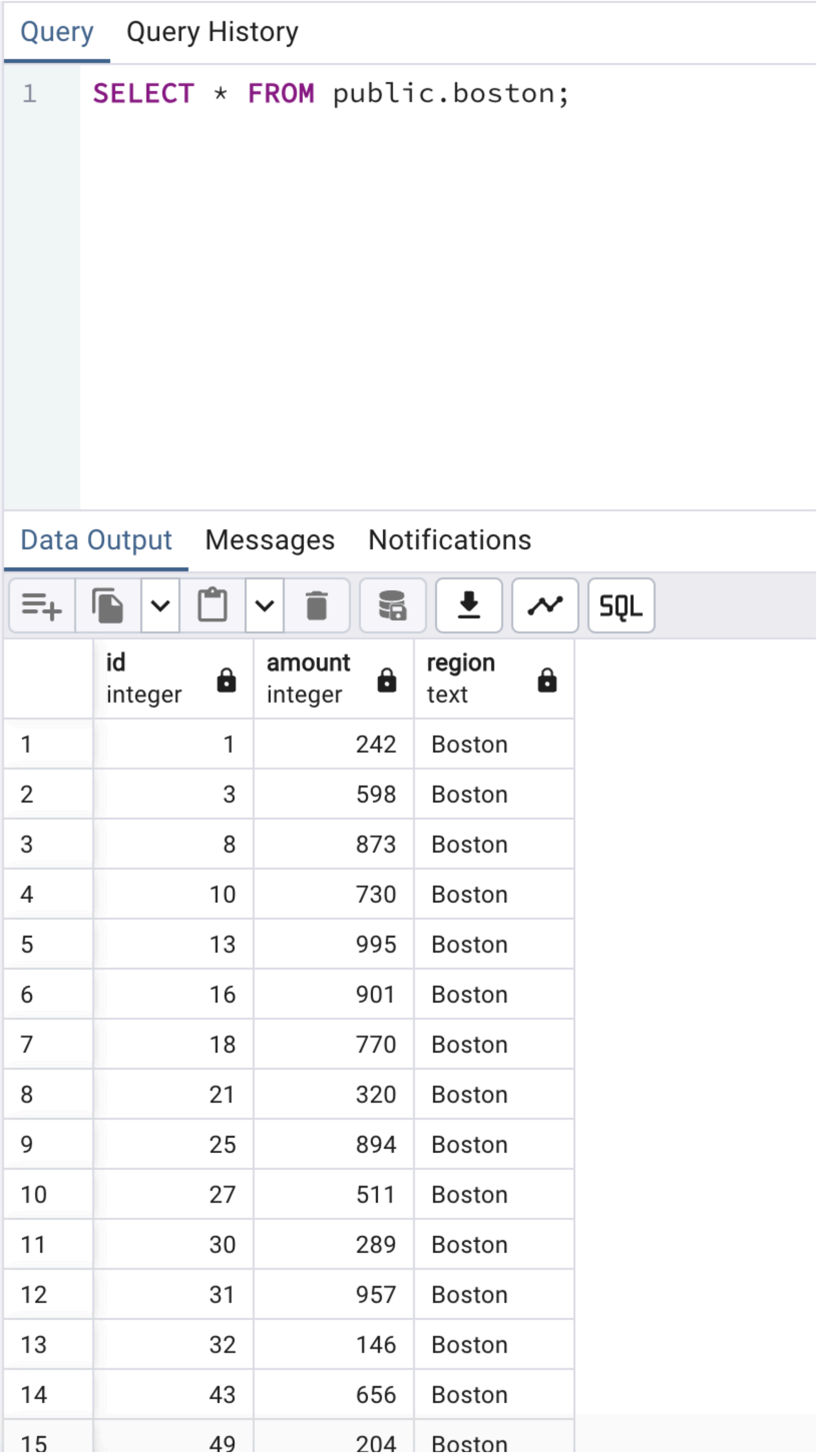
PgAdmin 4 Result Tables





Distributed Database Systems
CSE 512 – Fall 2024

Assignment 1

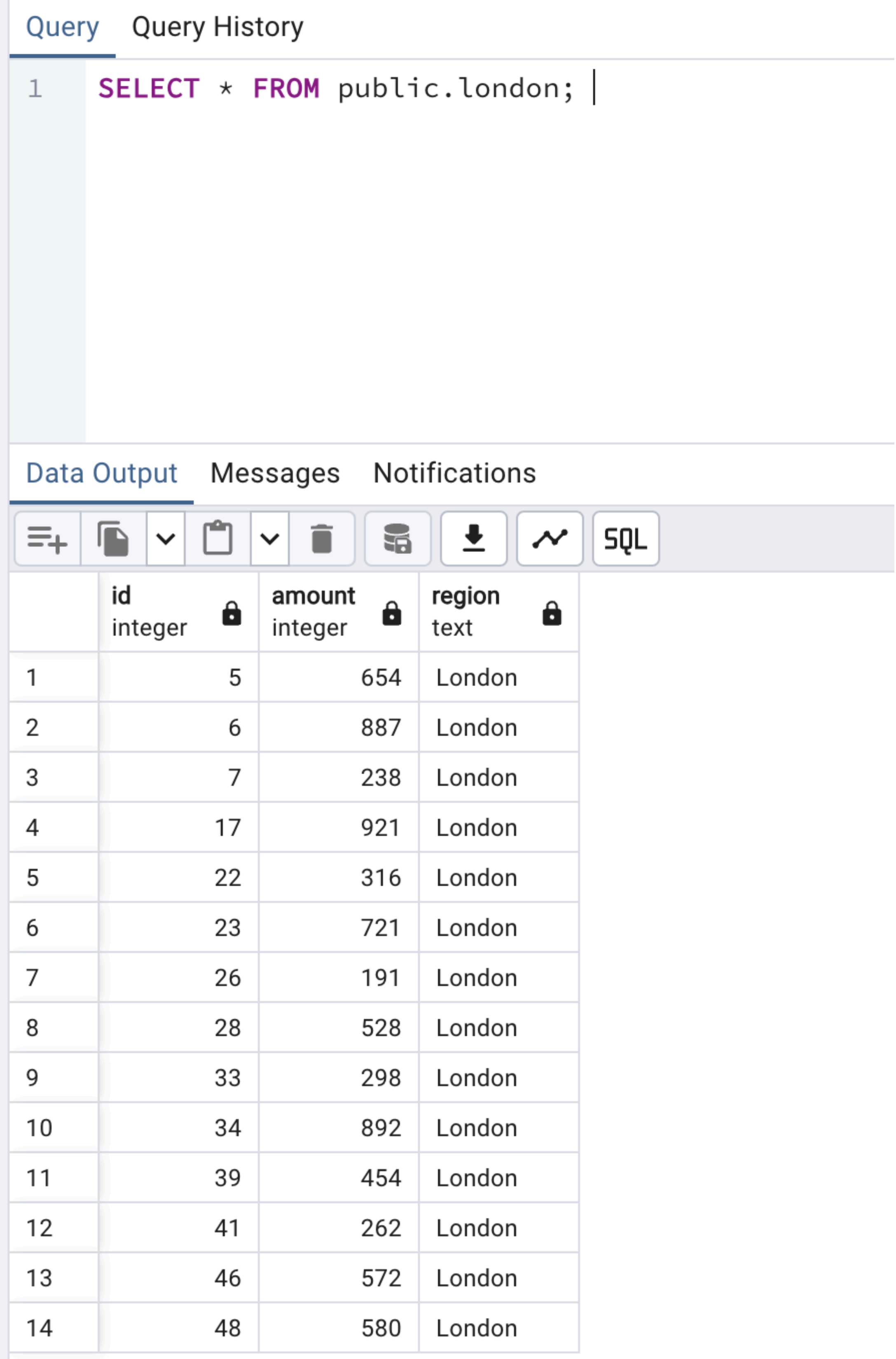


```
SELECT * FROM public.boston;
```

	id integer	amount integer	region text
1	1	242	Boston
2	3	598	Boston
3	8	873	Boston
4	10	730	Boston
5	13	995	Boston
6	16	901	Boston
7	18	770	Boston
8	21	320	Boston
9	25	894	Boston
10	27	511	Boston
11	30	289	Boston
12	31	957	Boston
13	32	146	Boston
14	43	656	Boston
15	49	204	Boston

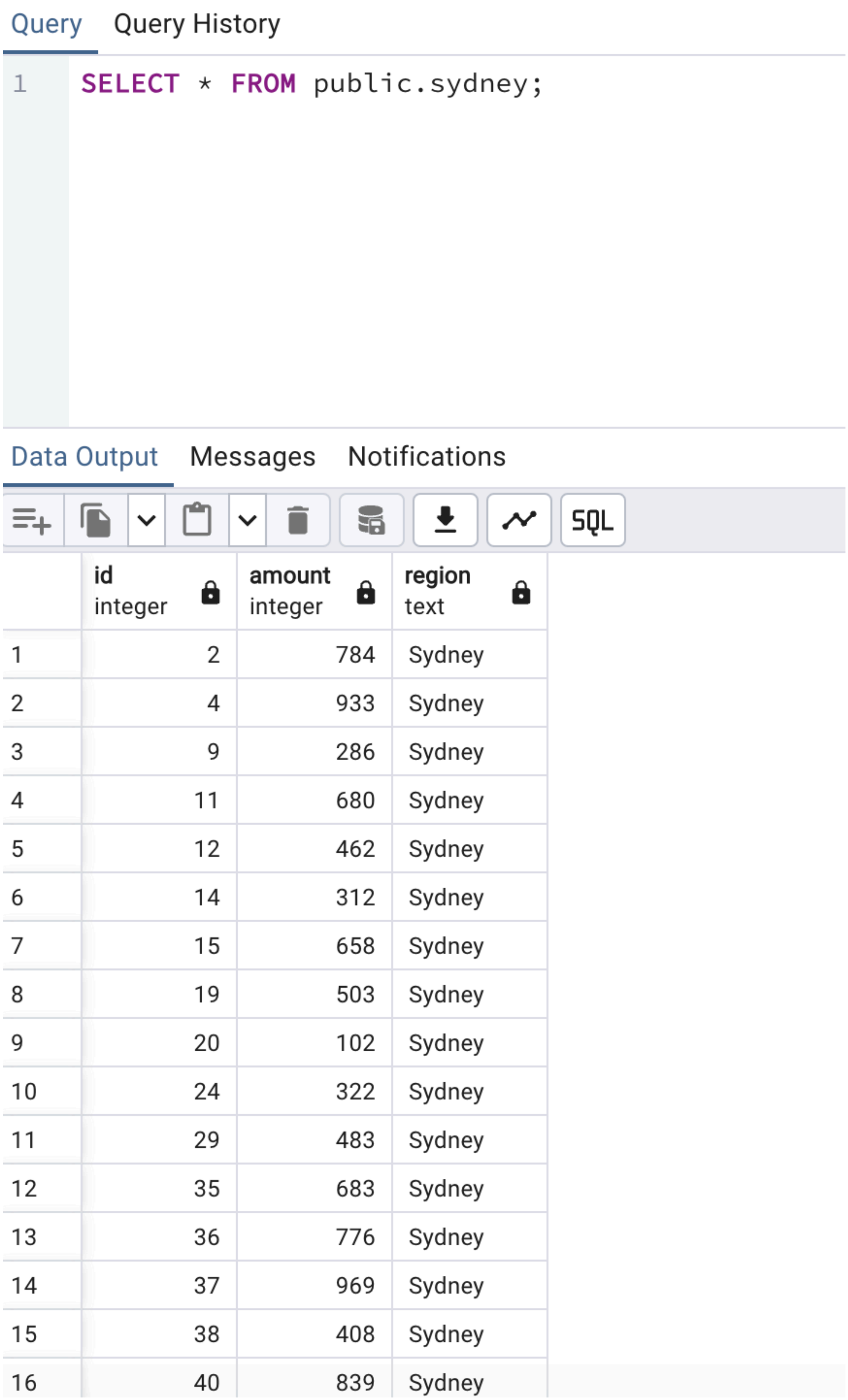
Distributed Database Systems
CSE 512 – Fall 2024

Assignment 1



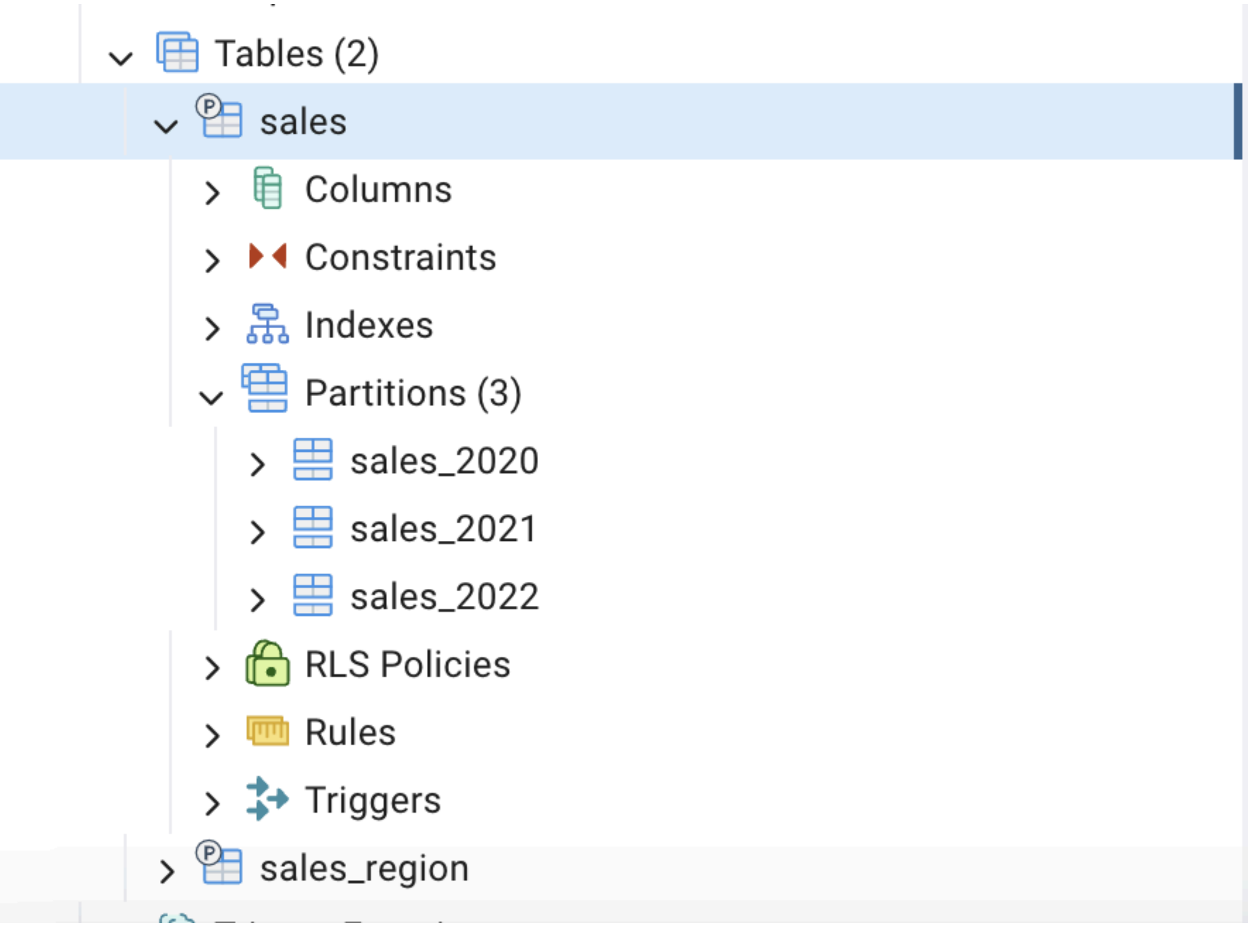
```
SELECT * FROM public.london;
```

	id integer	amount integer	region text
1	5	654	London
2	6	887	London
3	7	238	London
4	17	921	London
5	22	316	London
6	23	721	London
7	26	191	London
8	28	528	London
9	33	298	London
10	34	892	London
11	39	454	London
12	41	262	London
13	46	572	London
14	48	580	London



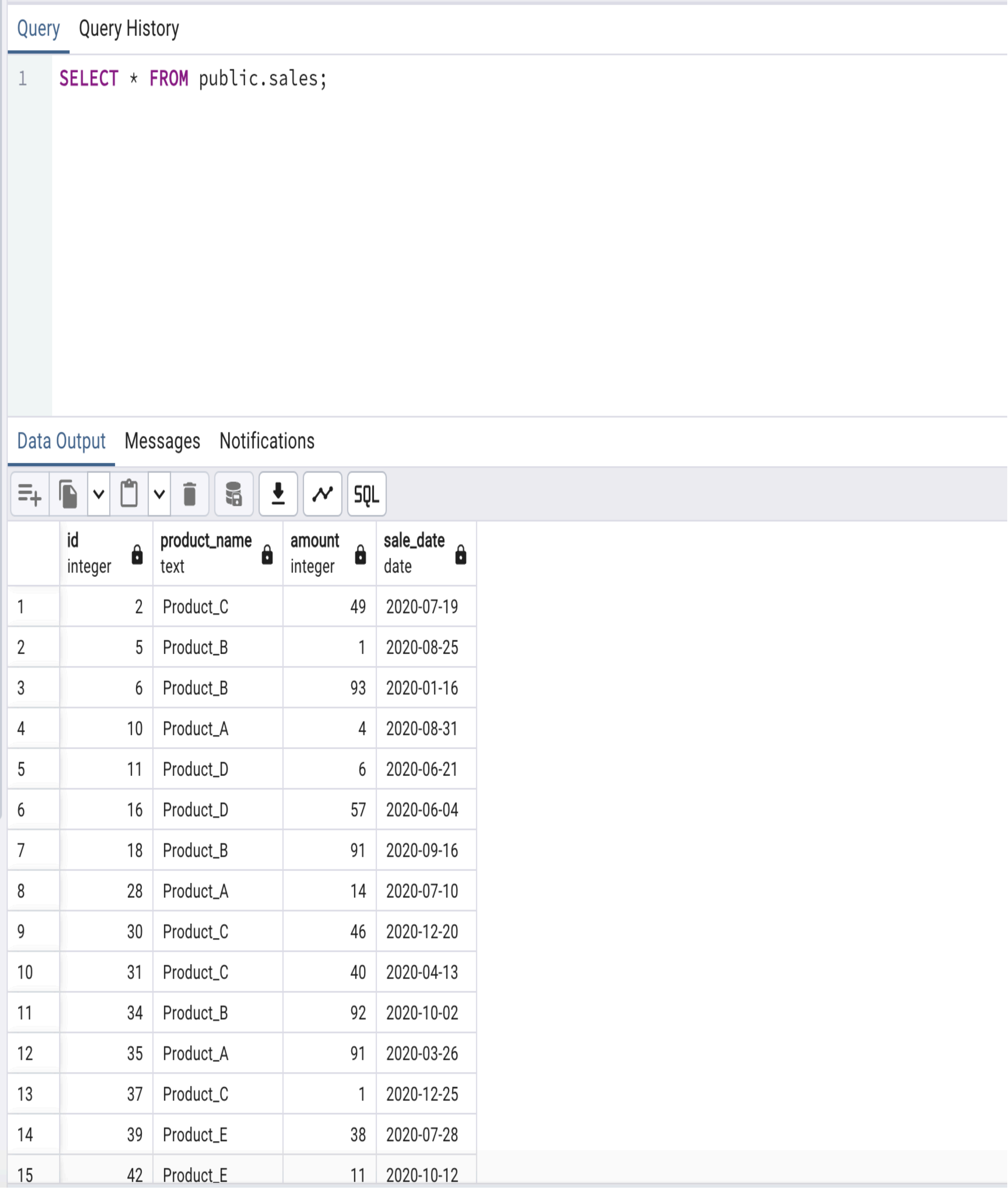
```
SELECT * FROM public.sydney;
```

	id integer	amount integer	region text
1	2	784	Sydney
2	4	933	Sydney
3	9	286	Sydney
4	11	680	Sydney
5	12	462	Sydney
6	14	312	Sydney
7	15	658	Sydney
8	19	503	Sydney
9	20	102	Sydney
10	24	322	Sydney
11	29	483	Sydney
12	35	683	Sydney
13	36	776	Sydney
14	37	969	Sydney
15	38	408	Sydney
16	40	839	Sydney


Tables (2)
sales
Columns
Constraints
Indexes
Partitions (3)
sales_2020
sales_2021
sales_2022
RLS Policies
Rules
Triggers
sales_region

Distributed Database Systems
CSE 512 – Fall 2024

Assignment 1

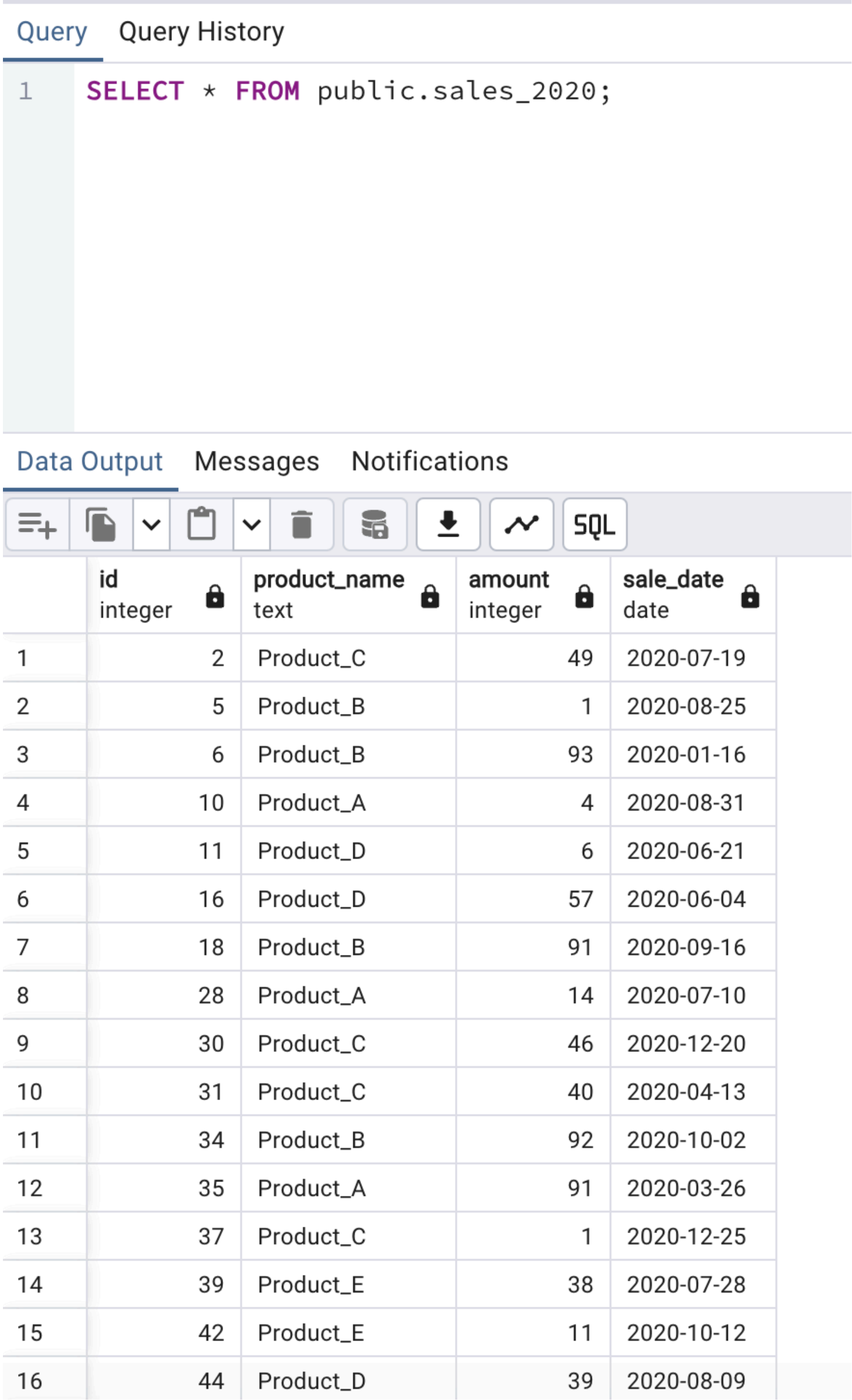


```sql
SELECT * FROM public.sales;
```

	id	product_name	amount	sale_date
	integer	text	integer	date
1	2	Product_C	49	2020-07-19
2	5	Product_B	1	2020-08-25
3	6	Product_B	93	2020-01-16
4	10	Product_A	4	2020-08-31
5	11	Product_D	6	2020-06-21
6	16	Product_D	57	2020-06-04
7	18	Product_B	91	2020-09-16
8	28	Product_A	14	2020-07-10
9	30	Product_C	46	2020-12-20
10	31	Product_C	40	2020-04-13
11	34	Product_B	92	2020-10-02
12	35	Product_A	91	2020-03-26
13	37	Product_C	1	2020-12-25
14	39	Product_E	38	2020-07-28
15	42	Product_E	11	2020-10-12

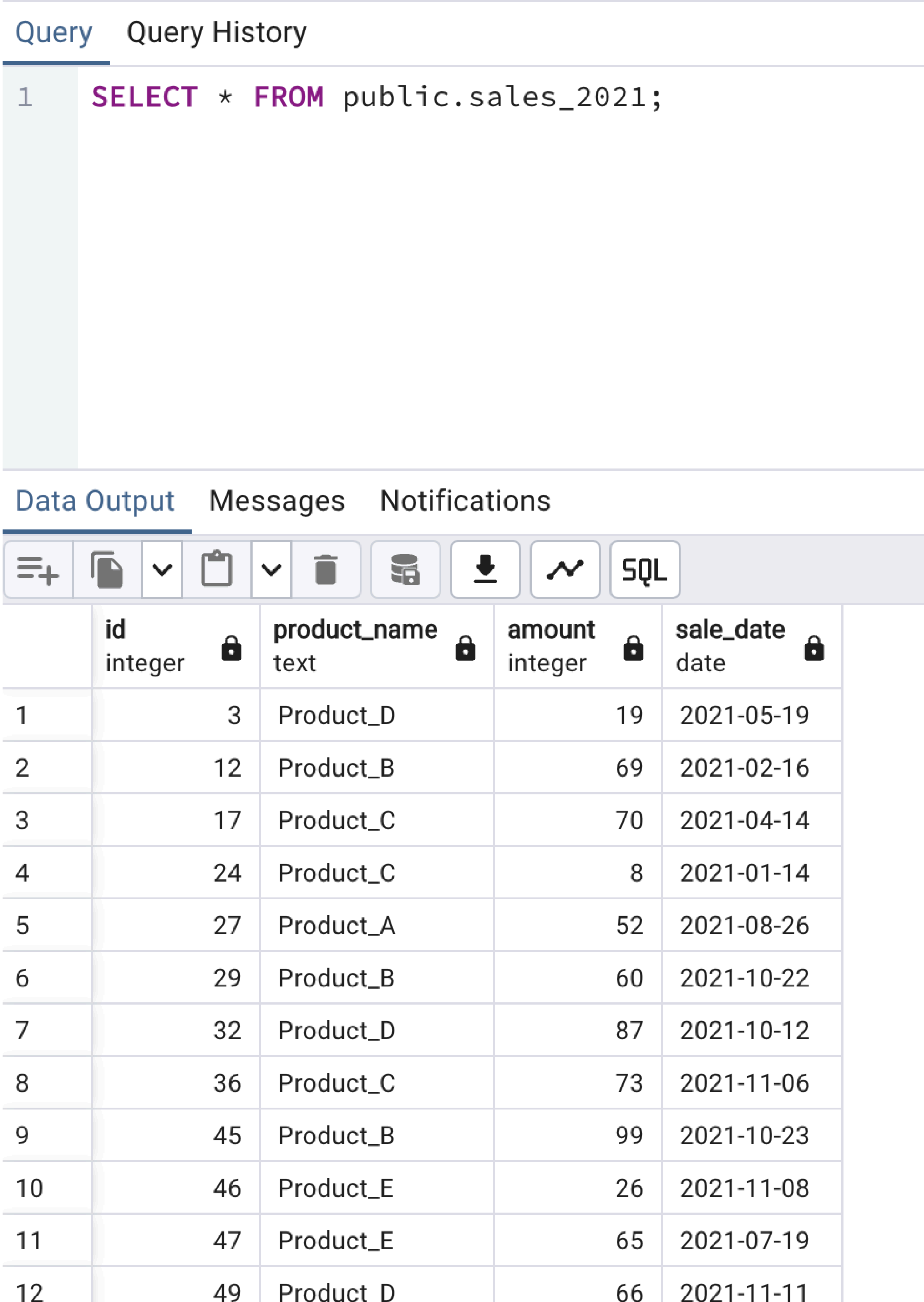
Distributed Database Systems
CSE 512 – Fall 2024

Assignment 1



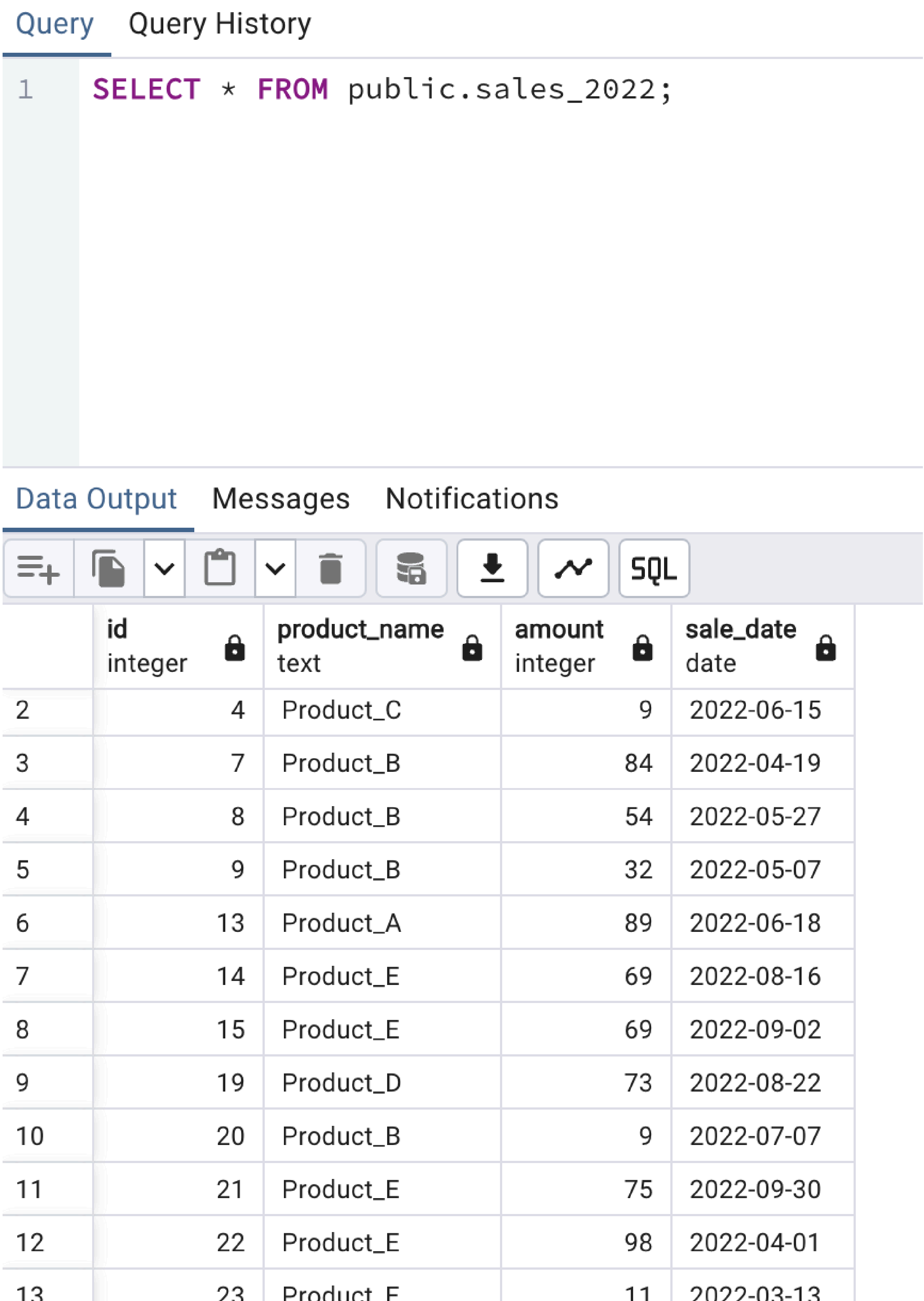
```
SELECT * FROM public.sales_2020;
```

	id integer	product_name text	amount integer	sale_date date
1	2	Product_C	49	2020-07-19
2	5	Product_B	1	2020-08-25
3	6	Product_B	93	2020-01-16
4	10	Product_A	4	2020-08-31
5	11	Product_D	6	2020-06-21
6	16	Product_D	57	2020-06-04
7	18	Product_B	91	2020-09-16
8	28	Product_A	14	2020-07-10
9	30	Product_C	46	2020-12-20
10	31	Product_C	40	2020-04-13
11	34	Product_B	92	2020-10-02
12	35	Product_A	91	2020-03-26
13	37	Product_C	1	2020-12-25
14	39	Product_E	38	2020-07-28
15	42	Product_E	11	2020-10-12
16	44	Product_D	39	2020-08-09



```sql
SELECT * FROM public.sales_2021;
```

	id integer	product_name text	amount integer	sale_date date
1	3	Product_D	19	2021-05-19
2	12	Product_B	69	2021-02-16
3	17	Product_C	70	2021-04-14
4	24	Product_C	8	2021-01-14
5	27	Product_A	52	2021-08-26
6	29	Product_B	60	2021-10-22
7	32	Product_D	87	2021-10-12
8	36	Product_C	73	2021-11-06
9	45	Product_B	99	2021-10-23
10	46	Product_E	26	2021-11-08
11	47	Product_E	65	2021-07-19
12	49	Product_D	66	2021-11-11



```
SELECT * FROM public.sales_2022;
```

	id integer	product_name text	amount integer	sale_date date
2	4	Product_C	9	2022-06-15
3	7	Product_B	84	2022-04-19
4	8	Product_B	54	2022-05-27
5	9	Product_B	32	2022-05-07
6	13	Product_A	89	2022-06-18
7	14	Product_E	69	2022-08-16
8	15	Product_E	69	2022-09-02
9	19	Product_D	73	2022-08-22
10	20	Product_B	9	2022-07-07
11	21	Product_E	75	2022-09-30
12	22	Product_E	98	2022-04-01
13	23	Product_E	11	2022-03-13