databricks ASP 1.1L - Explore Datasets Lab

Explore Datasets Lab

We will use tools introduced in this lesson to explore the datasets used in this course.

BedBricks Case Study

This course uses a case study that explores clickstream data for the online mattress retailer, BedBricks.
You are an analyst at BedBricks working with the following datasets: `events`, `sales`, `users`, and `products`.

Tasks

1. View data files in DBFS using magic commands
2. View data files in DBFS using dbutils
3. Create tables from files in DBFS
4. Execute SQL to answer questions on BedBricks datasets

```
%run ../../Includes/Classroom-Setup
```

```
Deleted the working directory dbfs:/user/odl_user_534131@databrickslabs.com/dbacademy/aspwd/asp_1_1l_explore_datasets_
lab


Your working directory is
dbfs:/user/odl_user_534131@databrickslabs.com/dbacademy/aspwd

The source for this dataset is
wasbs://courseware@dbacademy.blob.core.windows.net/apache-spark-programming-with-databricks/v02/

Skipping install of existing dataset to
dbfs:/user/odl_user_534131@databrickslabs.com/dbacademy/aspwd/datasets

Out[5]: DataFrame[key: string, value: string]
```

1. List data files in DBFS using magic commands

Use a magic command to display files located in the DBFS directory: **`dbfs:/databricks-datasets`**

You should see four items: **`events`**, **`products`**, **`sales`**, **`users`**

```
%fs ls dbfs:/databricks-datasets
```

	path	name	size
1	dbfs:/databricks-datasets/COVID/	COVID/	0
2	dbfs:/databricks-datasets/README.md	README.md	976
3			

4	dbfs:/databricks-datasets/SPARK_README.md	SPARK_README.md	3359
5	dbfs:/databricks-datasets/adult/	adult/	0
6	dbfs:/databricks-datasets/airlines/	airlines/	0
7	dbfs:/databricks-datasets/amazon/	amazon/	0

Showing all 51 rows.

2. List data files in DBFS using dbutils

- Use `dbutils` to get the files at the directory above and save it to the variable `files`
- Use the Databricks display() function to display the contents in `files`

You should see four items: `events`, `items`, `sales`, `users`

```
# TODO
files = dbutils.fs.ls('dbfs:/databricks-datasets')
display(files)
```

	path	name	size
1	dbfs:/databricks-datasets/COVID/	COVID/	0
2	dbfs:/databricks-datasets/README.md	README.md	976
3	dbfs:/databricks-datasets/Rdatasets/	Rdatasets/	0
4	dbfs:/databricks-datasets/SPARK_README.md	SPARK_README.md	3359
5	dbfs:/databricks-datasets/adult/	adult/	0
6	dbfs:/databricks-datasets/airlines/	airlines/	0
7	dbfs:/databricks-datasets/amazon/	amazon/	0

Showing all 51 rows.

3. Create tables below from files in DBFS

- Create the `users` table using the spark-context variable `c.users_path`
- Create the `sales` table using the spark-context variable `c.sales_path`
- Create the `products` table using the spark-context variable `c.products_path`
- Create the `events` table using the spark-context variable `c.events_path`

💡 Hint: We created the `events` table in the previous notebook but in a different database.

```
spark.sql(f"SET c.users_path = {users_path}")
spark.sql(f"SET c.sales_path = {sales_path}")
spark.sql(f"SET c.products_path = {products_path}")
spark.sql(f"SET c.events_path = {events_path}")
```

```
Out[7]: DataFrame[key: string, value: string]
```

```
%sql
CREATE TABLE IF NOT EXISTS users USING delta OPTIONS (path "${c.users_path}");
CREATE TABLE IF NOT EXISTS sales USING delta OPTIONS (path "${c.sales_path}");
CREATE TABLE IF NOT EXISTS products USING delta OPTIONS (path "${c.products_path}");
CREATE TABLE IF NOT EXISTS events USING delta OPTIONS (path "${c.events_path}");
```

OK

```
%sql
--CREATE TABLE IF NOT EXISTS events USING delta OPTIONS (path "${c.events_path}");
CREATE TABLE IF NOT EXISTS users USING delta OPTIONS (path "${c.users_path}");
CREATE TABLE IF NOT EXISTS sales USING delta OPTIONS (path "${c.sales_path}");
CREATE TABLE IF NOT EXISTS products USING delta OPTIONS (path "${c.products_path}");
```

```
OK
```

Use the data tab of the workspace UI to confirm your tables were created.

4. Execute SQL to explore BedBricks datasets

Run SQL queries on the `products`, `sales`, and `events` tables to answer the following questions.

- What products are available for purchase at BedBricks?
- What is the average purchase revenue for a transaction at BedBricks?
- What types of events are recorded on the BedBricks website?

The schema of the relevant dataset is provided for each question in the cells below.

4.1: What products are available for purchase at BedBricks?

The `products` dataset contains the ID, name, and price of products on the BedBricks retail site.

field	type	description
item_id	string	unique item identifier
name	string	item name in plain text
price	double	price of item

Execute a SQL query that selects all from the `products` table.

💡 You should see 12 products.

```
%sql
select distinct name from products
```

	name
1	Standard Full Mattress
2	Premium Queen Mattress
3	Premium Full Mattress
4	Standard Down Pillow
5	Premium Twin Mattress
6	Premium King Mattress
7	Standard King Mattress

Showing all 12 rows.

4.2: What is the average purchase revenue for a transaction at BedBricks?

The `sales` dataset contains order information representing successfully processed sales.
Most fields correspond directly with fields from the clickstream data associated with a sale finalization event.

field	type	description
order_id	long	unique identifier
email	string	the email address to which sales configuration was sent

field	type	description
transaction_timestamp	long	timestamp at which the order was processed, recorded in milliseconds since epoch
total_item_quantity	long	number of individual items in the order
purchase_revenue_in_usd	double	total revenue from order
unique_items	long	number of unique products in the order
items	array	provided as a list of JSON data, which is interpreted by Spark as an array of structs

Execute a SQL query that computes the average `purchase_revenue_in_usd` from the `sales` table.

The result should be `1042.79`.

```
%sql
select avg(purchase_revenue_in_usd) from sales
```

	avg(purchase_revenue_in_usd)
1	1042.7902657223433

Showing all 1 rows.

4.3: What types of events are recorded on the BedBricks website?

The `events` dataset contains two weeks worth of parsed JSON records, created by consuming updates to an operational database.
Records are received whenever: (1) a new user visits the site, (2) a user provides their email for the first time.

field	type	description
device	string	operating system of the user device
user_id	string	unique identifier for user/session
user_first_touch_timestamp	long	first time the user was seen in microseconds since epoch
traffic_source	string	referral source
geo (city, state)	struct	city and state information derived from IP address
event_timestamp	long	event time recorded as microseconds since epoch
event_previous_timestamp	long	time of previous event in microseconds since epoch
event_name	string	name of events as registered in clickstream tracker
items (item_id, item_name, price_in_usd, quantity, item_revenue in usd, coupon)	array	an array of structs for each unique item in the user's cart
ecommerce (total_item_quantity, unique_items, purchase_revenue_in_usd)	struct	purchase data (this field is only non-null in those events that correspond to a sales finalization)

Execute a SQL query that selects distinct values in `event_name` from the `events` table

You should see 23 distinct `event_name` values.

```
%sql
select distinct event_name from events
```

	event_name
1	mattresses
2	down
3	press
4	shipping_info
5	main
6	warranty
7	finalize

Showing all 23 rows.

Clean up classroom

```
classroom_cleanup()
```

```
Dropped the database dbacademy_odl_user_534131_databrickslabs_com_aspwd_asp_1_1l_explore_datasets_lab
```