INDEXING

METIS

Indexing

What is indexing in SQL?

Structure given to a table to help speed up queries and joins.

Index can be used to direct queries to a subset of data to find exact matches rather than scanning entire database.

Clustered

- Related to physical storage of data on your machine
- Only one allowed per table
- Tables without clustered index stored in unordered manner
- Typically reserved for Primary Key

Non-Clustered

- Not related to physical storage
- Structure not dictated by data rows
- Mapping from non-clustered index value to row locator
- Multiple allowed per table; values do not need to be keys

Indexing

What columns should be indexed?

1. Primary key (often as clustered index)
2. Foreign keys
3. Columns often used in WHERE or in JOIN ON clauses

What are the advantages and disadvantages to indexing?

Advantages

1. Speeds up queries and joins by creating computational efficiencies

Disadvantages

1. Must store indexes
2. Need to update indexes each time data is added or changed

Creating an Index

Inventory2 Table Creation

```
CREATE TABLE inventory2(
    id INT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    vendor_id INT FOREIGN KEY REFERENCES inv_vendors(id),
    quantity INT CHECK (quantity >= 0),
    upc_string VARCHAR(25) UNIQUE
);
```

```
CREATE INDEX IX_inventory2_vendorId ON inventory2(vendor_id);
```

Name the index

Note: CREATE INDEX yields a non-clustered index for specified column

Creating and Dropping an Index

	object_id	name	index_id	type_desc	is_unique	is_primary_key
1	910626287	PK__inventor__3213E83FAD6AE8E0	1	CLUSTERED	1	1
2	910626287	UQ__inventor__FD93C048A304DF83	2	NONCLUSTERED	1	0
3	910626287	IX_inventory2_vendorId	3	NONCLUSTERED	0	0

Dropping an index is also possible by referencing index name:

```
DROP INDEX IX_inventory2_vendorId ON inventory2;
```

Name of index