Lab 3: Creating Secondary Indexes (Cont.) Solution

CREATE INDEX Syntax:

```
CREATE [UNIQUE][BITMAP] INDEX  index_name
ON  table_name(column [, column]);
```

DROP INDEX Syntax:

```
DROP  INDEX  index_name;
```

Example:

- You want to improve the speed of query access to LAST_NAME column in the EMPLOYEES table:

```
CREATE  INDEX  emp_last_name_idx  ON  employees ( last_name)  ;
```

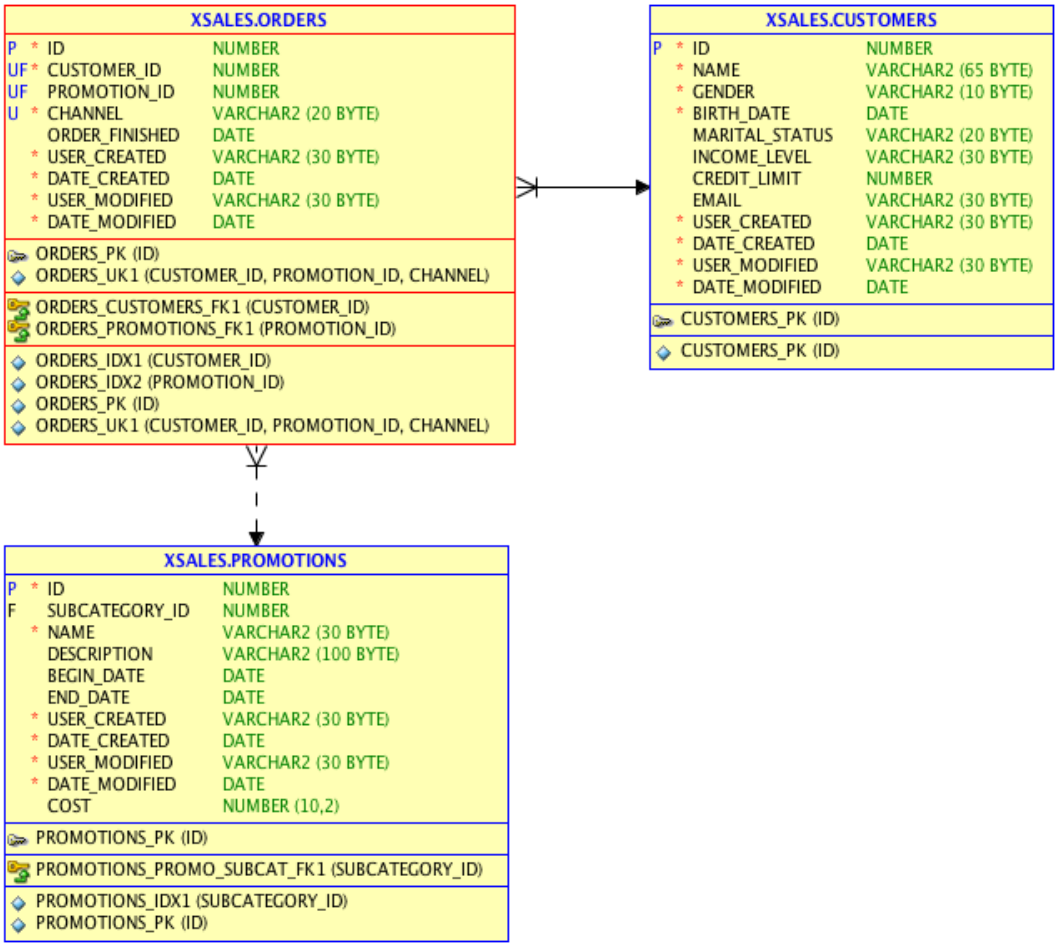
Order Index columns for Performance

- The order of columns in the CREATE INDEX statement can affect query performance. In general, specify the most frequently used columns first.
- If you create a single index across columns to speed up queries that access
  - For example, col1, col2, and col3;
  - Then queries that access just col1, or that access just col1 and col2, are also speeded up.
  - But a query that accessed just col2, just col3, or just col2 and col3 does not use the index.

Index Matching

- For index matching, you can use one or more operators such as >,>=,<,<=, LIKE, IN, IS NULL, BETWEEN…..AND in WHERE clause.
- Generally, search arguments in the WHERE clause such as "IS NOT NULL", "<>", "!=", "!>", "!<", "NOT", "NOT EXISTS", "NOT IN", "NOT LIKE", and "LIKE '% 500'" prevents Oracle from using an index to perform the search (*however, not always*).

Practices



1. View the existing indexes of the XSALES.CUSTOMERS, XSALES.ORDERS and XSALES.PROMOTIONS tables.

Table Name	Index Name (s)	Index Type (s) [B-tree vs Bitmap]	Column Name (s)
XSALES.CUSTOMERS	CUSTOMERS_PK	B-Tree	ID
XSALES.ORDERS	ORDERS_PK	B-Tree	ID
	ORDERS_UK1	B-Tree	CUSTOMER_ID, PROMOTION_ID, CHANNEL
	ORDERS_IDX1	B-Tree	CUSTOMER_ID
	ORDERS_IDX2	B-Tree	PROMOTION_ID
XSALES.PROMOTIONS	PROMOTIONS_PK	B-Tree	ID
	PROMOTION_IDX1	B-Tree	SUBCATEGORY_ID

2. Write SQL statements to query information in the following, Please find out the use of the indexes found in Question 1 in each query.

   2.1. List customer name, marital status, income level and birth date of customer named 'Antony Chinn'.

```
select name, marital_status, income_level, birth_date
from xsales.customers
where name = 'Antony Chinn';
```

   2.2. List order id, promotion id and channel of the customer number 14053.

```
select id, promotion_id, channel
from xsales.orders
where customer_id = 14053;
```

   2.3. List order id, promotion id and channel of all customers whose customer number is between 10000 and 15000 and has orders with promotion id 999 and ordered from the Internet channel.

```
select id, promotion_id, channel
from xsales.orders
where customer_id between 10000 and 15000
and promotion_id = 999
and channel = 'Internet';
```

   2.4. List customer name, marital status and order finished date of all customers whose marital statuses are 'single' and credit limits are 15000.

```
select name, marital_status, order_finished
from xsales.customers c join xsales.orders o
on c.id = o.customer_id
where marital_status='single'
and credit_limit = 15000;
```

   2.5. Find promotion name and the number of orders in each promotion.

```
select p.name, count(o.id) as orders#
from xsales.orders o join xsales.promotions p
on o.promotion_id = p.id
group by p.name;
```

3. Create your own questions and answer those by writing SQL queries that has the use of the index(s) found in Question 1. Do 5 questions with different usage of the indexes.