Takeaway Sheet 1 (Lectures 1-3)

L1.1

Example: Rock Climbing Log for Washington State

- **Elements**: Climbing routes.
- Attributes: Route, Difficulty, Type, Date.
    - Route (String): Name of the climb.
    - Difficulty (String): Rating, e.g., "5.10b".
    - Type (String): Climbing style, e.g., "Sport".
    - Date (Date): When climbed.
- Example: "Liberty Bell", "5.11a", "Trad", 2020-09-20.

In the Rock Climbing Log example:

- Elements (Rows): Individual climbs.
- Attributes (Fields): Route (String), Difficulty (String), Type (String), Date (Date).
- Types and Domains: Attributes are Strings and Date, with domains being route names, difficulty ratings, climbing types, and specific dates, respectively.

L1.2

ID is the most suitable key due to its inherent uniqueness, but given unique values in the data, URL and NumVisits could also function as keys.

L2.1

- Alternative name for "row": A row can alternatively be called Tuple or Record.
- If the table is an array of Java objects, what is printed to the screen: Assuming each row is represented as an instance of a Java object, printing the array to the screen would likely display references to these object instances. To print specific data, you would need to iterate over the array and access each object's attributes, printing them as needed.

L2.2

To find all the jobs with a salary greater than or equal to 60,000, the SQL query would be:

```sql
SELECT * FROM jobs WHERE salary >= 60000;
```

(Note: This assumes the existence of a `jobs` table with a `salary` column.)

L2.3

- A foreign key can be the primary key in the referred table: True.
- A foreign key can be a "normal column" (i.e., not a primary key) in the referred table: True.

L3.1

Pseudocode:

1. Initialize three pointers, i, j, and k to 0. i for array1, j for array2, and k for the mergedArray.
2. While i < length of array1 AND j < length of array2, do: a. If array1[i] < array2[j], assign array1[i] to mergedArray[k], increment i and k. b. Else, assign array2[j] to mergedArray[k], increment j and k.
3. While i < length of array1, copy remaining elements of array1 to mergedArray, increment i and k.
4. While j < length of array2, copy remaining elements of array2 to mergedArray, increment j and k.
5. Output mergedArray as the combined sorted array.

L3.2

To find the names of everyone who drives a Civic or a Ferrari, assuming a `drivers` table with `name` and `car` columns, the query could look like this:

```sql
SELECT name FROM drivers WHERE car IN ('Civic', 'Ferrari');
```