**[WHW1] Hiring new software engineers.**

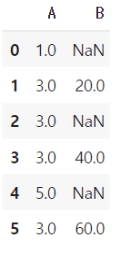
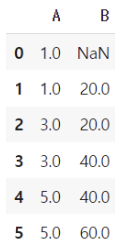
1. What kind of data would be necessary?

Educational background, Technical skills, Project performance metrics, co-worker’s feedback, Career progression

1. Assuming Employee table is one such data, list 3 features/attributes.

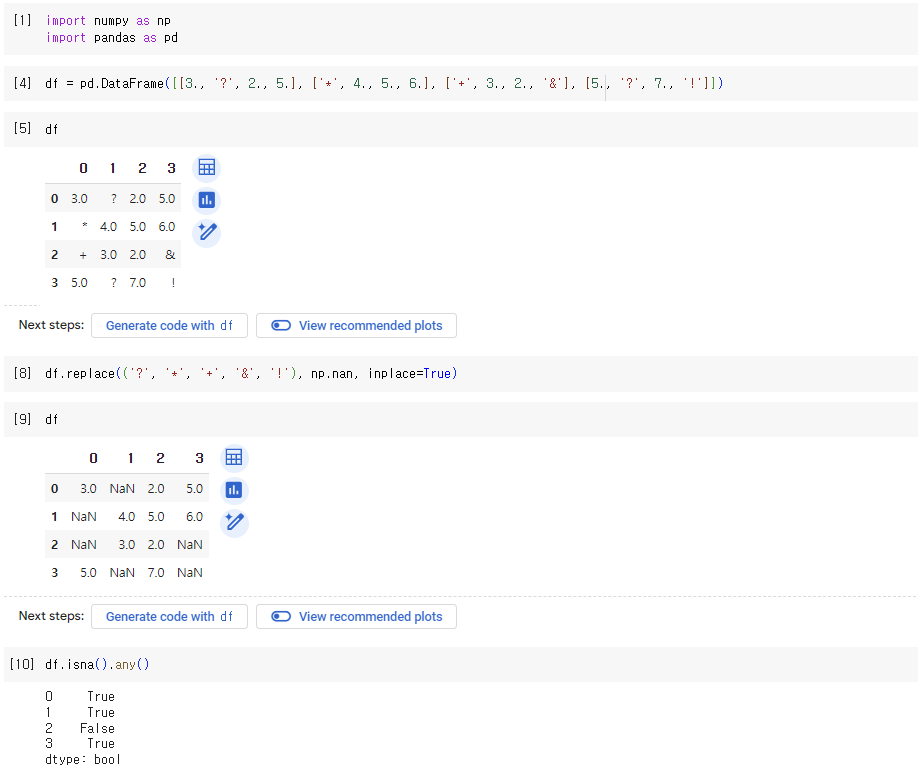
|  |  |  |
| --- | --- | --- |
| Technical skills | Performance quality | Tenure |
| **. . .** | **. . .** | **. . .** |

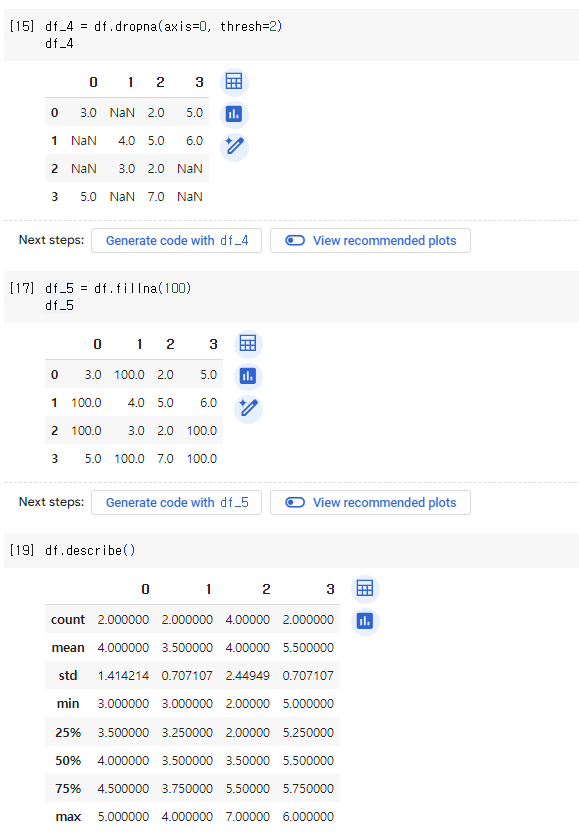
**[WHW2] ffill & bfill**

1. What if the first row of last row contains NaN?
2. First row, using ffill -> Still NaN **→**
3. Last row, using bfill -> Still NaN
4. What if the replacement row also contains NaN?

Replace NaN data using other data’s mean, median, mode value.

**[PHW, 47pg]**







**[WHW, 49pg] Wrong data**

1. Difference of unique , primary key

Unique key doesn’t necessarily identify each row uniquely, and it allows NULL value.

However, Primary key uniquely identifies in a table, and not allow NULL.

1. ACID, transaction?

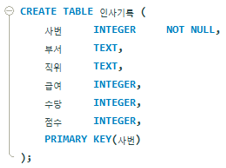
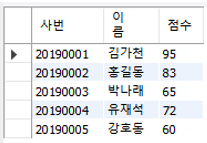
ACID(Atomicity, Consistency, Isolation, Durability) : Essential characteristics of transaction in DB system.

Transaction: Logical unit of work performed on DB. (DB operations, such as inserts, updates, deletes, of queries)

**[WHW, 54pg]**

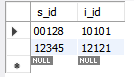
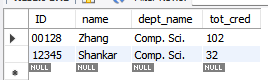
1. Describe various RDBM integrity constraints.

**Primary key constraint**: Each row in a table is uniquely identifiable by specifying all columns as the primary key. (Uniqueness, Non-nullability)

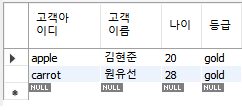
**Foreign key constraint** : Relationship between two tables by specifying that the values in column in one table must match the values in primary/unique key column in another table.

Ex) ID <- s\_id (foreign reference)



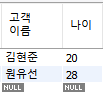
**Unique constraint** : The values in columns are unique across all rows in a table (allow NULL value)

Ex) 고객아이디 : unique



**Check constraint** : Condition that must be satisfied for the data in a column. (insert, update column based on the defined condition)

Ex) 나이: always greater than zero, can’t too much big number



1. Describe ACID problems.

**Lost update** : When two transactions concurrently read the same data, and one transaction updates data before the other transaction completes its update, causing the changes made by the first transaction to be lost.

Ex) “Accounts” table, Transaction1=A transfers $100 to B’s account, Transaction2=C transfers $50 to D’s account.

* If transaction1 updates A’s account before transaction2 completes its update, the changes made by transaction2 may be lost.

**Dirty read** : When one transaction reads data that has been modified by another transaction that has not yet been committed. If second transaction is rolled back, first transaction become “dirty”.

Ex) Transaction1=User updates email address in table to “user@naver.com”

Transaction2=While transaction1 still in progress, reads the updated email address.

* If transaction1 is later rolled back (some error), transaction2 read dirty value

**Unrepeatable read** : When transaction reads the same data multiple times, but the data changes between each read due to other concurrent transactions committing their changes.

Ex) transaction1=User A reads bank account, $500, transaction2=Another user transfers $200 from A’s account, committing the change. Again transaction1, A reads bank account, which is now $300.

* The value read in the second read is different from the first read, causing inconsistency and unpredictability.

**Lost transaction** : When transaction completes all its operations, but fails to commit, resulting in the loss of its changes without any rollback.

Ex) User transfers $100 to friend’s account. All updates are made successfully, but before transaction committed, a power off.

* The changes made by the transaction are lost, the money is neither in User’s account nor in friend’s account.