1. Command that automatically generates sequential numbers: SEQUENCE
2. Generates a numeric value: NEXTVAL or CURRVAL
3. Returns the next available sequence value: NEXTVAL
4. Specifies the interval between sequence numbers: INCREMENT BY
5. Specifies a maximum value of 102710^{27}1027 for an ascending sequence and -1 for a descending sequence (default): MAXVALUE
6. Returns the current sequence value: CURRVAL
7. Specifies the minimum sequence value: MINVALUE
8. Specifies whether the sequence continues after reaching its max/min values: CYCLE/NOCYCLE
9. Specifies a minimum value of 1 for ascending sequence and −1026-10^{26}−1026 for descending sequence (default): MINVALUE
10. Specifies a maximum or default value the sequence can generate: MAXVALUE
11. Specifies the first sequence number to be generated: START WITH
12. Specifies how many values the server pre-allocates and keeps in memory: CACHE/NOCACHE

1. CREATE TABLE seq_d_songs AS
   SELECT * FROM d_songs;
2. CREATE SEQUENCE seq_d_songs_seq
   START WITH 100
   INCREMENT BY 2
   MAXVALUE 1000
   NOCACHE
   NOCYCLE;
3. SELECT sequence_name, min_value, max_value, increment_by, cache_size, cycle_flag
   FROM user_sequences
   WHERE sequence_name = 'SEQ_D_SONGS_SEQ';
4. INSERT INTO seq_d_songs (id, title, duration, artist, type_code)
   VALUES (seq_d_songs_seq.NEXTVAL, 'Island Fever', 5, 'Hawaiian Islanders', 12);

   INSERT INTO seq_d_songs (id, title, duration, artist, type_code)
   VALUES (seq_d_songs_seq.NEXTVAL, 'Castle of Dreams', 4, 'The Wanderers', 77);
5. SELECT seq_d_songs_seq.CURRVAL FROM dual;
6. Three benefits of using sequences are is ensures uniqueness, automates numbers generation, and improves efficiency
7. The advantages of caching sequence values are it speeds up insert operations and minimizes contention
8. Three reason why a gap may occur in a sequence are transaction rollbacks, a system crash, or a manual increment